NPL # N

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 February 2001 (01.02.2001)

PCT

(10) International Publication Number
WO 01/007093 A1

(51) International Patent Classification[7]: A61L 9/00, 9/03, 9/12

(21) International Application Number: PCT/US00/20029

(22) International Filing Date: 21 July 2000 (21.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/145,412	23 July 1999 (23.07.1999)	US
60/155,126	22 September 1999 (22.09.1999)	US
60/158,495	8 October 1999 (08.10.1999)	US
60/158,615	8 October 1999 (08.10.1999)	US
60/181,113	8 February 2000 (08.02.2000)	US
60/181,115	8 February 2000 (08.02.2000)	US
60/184,809	24 February 2000 (24.02.2000)	US
60/188,332	9 March 2000 (09.03.2000)	US

(71) Applicant *(for all designated States except US)*: **LIBRA DIGITAL, LLC** [US/US]; 1814 Franklin Str., Suite 720, Oakland, CA 94612 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BELLENSON, Joel** [US/US]; 244 Lakeside Drive, Apt. 15, Oakland, CA 94612 (US). **SMITH, Dexster** [US/US]; 868 Trestle Glen Road, Oakland, CA 94610 (US). **KERR, Jeff** [US/US]; 1713 Alameda de las Pulgas, Redwood City, CA 94025 (US). **HUNICKE-SMITH, Scott** [US/US]; 58 Henderson Place, Menlo Park, CA 94025 (US).

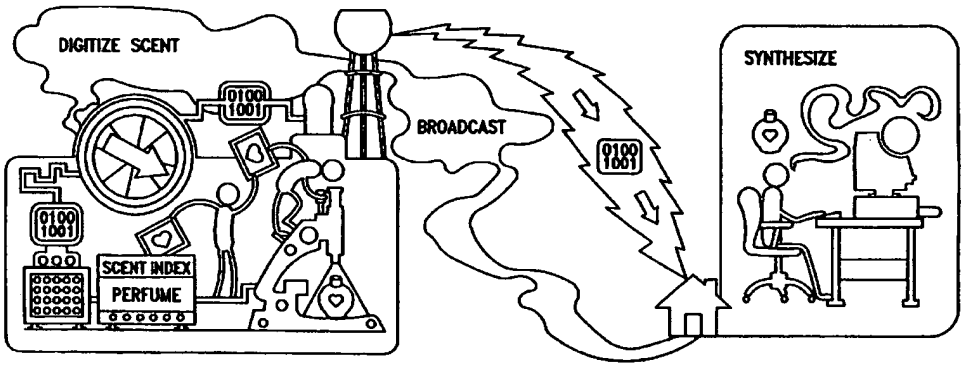
(74) Agents: **CERPA, Robert, K.** et al.; Morrison & Foerster LLP, 755 Page Mill Road, Palo Alto, CA 94304-1018 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

[Continued on next page]

(54) Title: ELECTRONIC RECORDING, ANALYSIS, EDITING, AND PLAYBACK OF SCENTS



(57) Abstract: A system for recording, analyzing, editing, and playback of scents is disclosed. A scent can be sampled and represented in various manners, such as by reference to an activation pattern of olfactory receptors (determined experimentally or by using computer programs), description of chemical structure, description of its characteristics by a panel of human subjects, or its pattern of interaction with an artificial scent-sensing device (an "artificial nose" device). A scent emitter having a set of chemicals which can be volatilized is disclosed for re-creation of the scents from the stored scent representation. The scent representation can be transmitted over computer networks, such as the Internet, for re-creation at remote locations, and can be stored for re-creation of the scent at times different from the time of sampling of the scent. The system finds applications in advertisements which may appear on pages on the World Wide Web, in re-creating scents for forensic or quality control analysis, as enhancements to virtual reality simulations, customs creation of scents by scent designers, and other uses.

WO 01/007093 A1

IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(48) Date of publication of this corrected version:
26 September 2002

(15) Information about Correction:
see PCT Gazette No. 39/2002 of 26 September 2002, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR ELECTRONIC RECORDING, ANALYSIS, EDITING, AND PLAYBACK OF SCENTS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority benefit of co-pending United States Provisional Patent Applications Serial Nos. 60/145,412, filed July 23, 1999; 60/155,126, filed September 22, 1999; 60/158,495, filed October 8, 1999; 60/158,615, filed October 8, 1999; 60/181,113, filed February 8, 2000; 60/181,115, filed February 8, 2000; 60/184,809, filed February 24, 2000; and 60/188,332, filed March 9, 2000. This application is also related to U.S. Nonprovisional Patent Application Serial No. (Not Yet Assigned), filed July 21, 2000, and entitled "System and Method For Electronic Recording, Analysis, Editing, and Playback of Scents" (Attorney Docket No. 422852000340). The contents of those applications are hereby incorporated by reference herein in their entirety.

MICROFICHE/COPYRIGHT REFERENCE

A Microfiche Appendix is included in this application that contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the Microfiche Appendix, as it appears in the U.S. Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

STATEMENT OF RIGHTS TO INVENTIONS MADE UNDER FEDERALLY SPONSORED RESEARCH

Not applicable.

TECHNICAL FIELD

The present invention is directed to a system and method for electronically recording a scent, optionally transmitting the electronic representation of the scent over a network, analyzing and editing the scent in electronic form, and re-creating the scent from the electronic representation. The present invention allows archival storage of scents and re-creating of

scents on-demand either locally or at a remote site. Many practical applications can take advantage of this additional qualitative dimension.

BACKGROUND OF THE INVENTION

It is well-known that the sense of smell is important in human perception of events, objects, and the surrounding environment. Scents from foods and the environment dynamically change an individual's interaction and in effect make each experience unique. Common interactions throughout the day involve scents from various sources. City sidewalks, coffee shops, flower shops, and many other sources of smell illustrate the blending of everyday olfactory sensory interaction between the people and the environment.

Many previous attempts have been made to enrich people's experience in a variety of settings by the use of scents. For example, there have been efforts to incorporate scents to various presentations by spreading odors in auditoriums or viewing rooms through the use of heating and spraying methods, with the purpose of enhancing cinematic, television, or other video entertainment. Additionally, other methods to provide odor have included items such as pre-programmed cartridges or other containers containing odorous substances. For example, U.S. Patent No. 5,591,409 and U.S. Patent No. 5,724,256 provide example of how odorant containers allow scent particles to permeate the air to provide the desired aroma. However, these patents do not disclose how the scents are to be classified and indexed.

Additional scent-related technology includes that described in US Patent No. 6,053,738, which is directed to a "sense-simile transmission machine;" US Patent No. 6,024,783, which describes aroma sensory stimulation in multimedia; US Patent No. 6,004,516 and International Patent Application WO 99/08174, which describe an apparatus for generating odor upon electronic signal demand; US Patent No. 5,999,105, which describes a multiple sensory message center apparatus; US Patent No. 5,996,396, which describes an apparatus for determining odor levels in gas streams; US Patent No. 5,995,770, which describes a system and method for delivering a scent to a photographic print; US Patent No. 5,987,466, which describes presenting web pages with discrete, browser-controlled complexity levels; US Patent No. 5,985,214, directed to a system and methods for rapidly identifying useful chemicals in liquid samples; US Patent Nos. 5,974,444, 5,696,901, and 5,544,320, which describe a remote information service access system based on a client-server-service model; US Patent No. 5,972,290, which describes a process and equipment for the programmed scenting of environments; US Patent No. 5,966,126, to a graphic user interface for database systems; US Patent Nos. 5,963,302 and 5,832,320, which describes a

process and device for diffusing perfumes that correspond to events or scenes during cinematographic representations and the like; US Patent No. 5,949,522, which describes a multimedia linked scent delivery system; US Patent No. 5,930,803, to a method, system and computer program for visualizing an evidence classifier; US Patent No. 5,887,118, which describes an olfactory card; US Patent No. 5,782,692, to a time-segmented multimedia game playing and authoring system; US Patent Nos. 5,767,385 and 5,613,909, directed to an automated forced-choice dynamic dilution olfactometer and method of operating the same; US Patent No. 5,761,071, which describes a browser/kiosk system; US Patent No. 5,727,186, a simulation apparatus and gas dispensing device used in conjunction therewith; US Patent No. 5,724,256, directed to a computer controlled olfactory mixer and dispenser for use in multimedia computer applications; US Patent No. 5,682,330, to a repetitive event analysis system; US Patent No. 5,591,409, which describes a method for providing aromas; US Patent No. 5,574,821, to a plug-in volatile substance dispenser and a method for dispensing volatiles; US Patent No. 5,526,281, which describes a machine-learning approach to modeling biological activity for molecular design and to modeling other characteristics; US Patent No. 5,382,410, directed to an electrostatic vapor/aerosol generator with method and apparatus for conditioning building spaces; US Patent No. 5,196,171, an electrostatic vapor/aerosol/air ion generator; US Patent No. 5,120,709, a method for enhancing fragrance applications; US Patent No. 5,090,232, a method and apparatus for detecting odors; US Patent No. 5,071,621, a method of supplying scents to a room of a motor vehicle; US Patent No. 5,069,877, an article for diffusing volatile substances, particularly perfume; US Patent No. 4,934,386, an apparatus for assessing responses of humans to stimulants; US Patent No. 4,629,604, which describes a multi-aroma cartridge player; US Patent No. 4,611,294, a method of and apparatus for monitoring odorizer performance; US Patent No. 4,532,530, a bubble jet printing device; US Patent No. 4,037,352, an electrical device which emits insecticidal vapors; US Patent No. 4,009,384, a lamp scent unit; US Patent No. 2,871,526, directed to porous containers for thickened air odor control liquids; and US Patent No. 2,562,959, electromechanical scent distribution to accompany a motion picture.

Other documents describing scent-related technology include WO 00/15269, to methods and apparatus for odor reproduction, and WO 00/15268, methods and apparatus for odor transmission; WO 99/48539, which describes a method for aroma delivery; WO 99/48469, directed to a composition for aroma delivery; WO 99/38102 A1, which describes an olfactory emitting system; WO 99/16476, to a device for reproducing or synthesizing odors or aromas; WO 96/02887, a virtual reality and remote reality system; and

WO 90/05965, a process for identifying a person for access to an installation, premises and/or equipment, on the basis of the person's characteristic odor; EP 0 883 049 , to an olfactometer; EP 0 831 384, to methods and systems for controlling olfactory stimuli; EP 0 775 961, to a virtual reality and remote reality system; EP 0 632 268, to an apparatus and method for determining the indoor air quality within an enclosed space; EP 0 411 681, to cooking oils with reduced odor-producing tendencies; EP 0 325 468, to a method of supplying scents to a room of a motor car; EP 0 290 065 B1, which describes food or beverage compositions with altered flavor display; EP 0 123 746 A2, which describes a disc-playing aroma generator; CN 1087994, which describes a complex array piezoelectric crystal olfaction biological sensing detector; DE 196 04 600, which describes a procedure and mechanism for the detecting and reduction of smells and odors; FR 2,501,468, which describes an insecticide evaporator casing; and GB 2,279,010, to a device for selectively generating perfume gas.

It would be of great utility to be able to record any scent in electronic format for storage, transmission, analysis, and playback. Much like the use of recorded audio and video, recordings of various scents could be used in a variety of contexts to enrich the experience of simulation. These scents, indexed by various categories, could be retrieved for transmission and playback within the proper context. Expanded possibilities for commercial, entertainment, administrative purposes exist with such capabilities. It would be highly desirable to employ a system capable of capturing, recording, transmitting, reconstructing, and emitting the necessary scents to enhance various multimedia presentations. Provided that the scents are an accurate representation or approximation of the desired aromas, the featured presentations would have an added dimension of reality.

SUMMARY OF THE INVENTION

In one embodiment, the invention embraces a system and method for electronic recording, analysis, editing, and playback of scents is described. There are two basic subsystems in this embodiment: (i) a recording subsystem for capturing and interpreting scents as well as converting scents to electronic files and analyzing or editing those files electronically, and (ii) an emission subsystem for reproducing scents. The present invention will make possible the capturing, recording, transmitting, reconstructing, and emitting of any scent, or of a particular subset of scents, for a wide variety of applications. The scents will be electronically recorded and indexed for proper retrieval.

In this embodiment, scents can be captured, analyzed and recorded by a sensory device using various methods. Scent capture can be initiated by the user or by an automatic sensing system. A scent can be analyzed in terms of its interaction with olfactory neurons of a mammalian, preferably human, olfactory system (measured either by interaction with a biological preparation of receptors or computer simulation of the interaction with receptors), or in terms of its perception by a panel of mammalian, preferably human, subjects, or in terms of the structure of the chemical components of the scent itself. The interaction with olfactory neurons can be determined experimentally, in vitro, by determining the interaction of an odorant with olfactory neurons of a given type (i.e., an olfactory neuron expressing a particular olfactory receptor). Alternatively, the interaction with olfactory neurons can be determined using a computer simulation which provides information regarding interaction of an odorant with the olfactory receptors. A panel of subjects can be used to represent odors in terms of their perception. The data so generated can be used to represent a scent in a manner which can be recorded in electronic or other format, stored in media such as computer memory, disks, or printed format, and transmitted over a data network. The representation of the scent can be used to re-create the scent at a local or remote site using an emitter module. The representation of the scent allows for scent editing, where desirable aspects of an odor are enhanced or added and undesirable aspects are attenuated or eliminated.

The various embodiments of the present invention will be useful for various commercial purposes. A particular scent could be used in conjunction with a product to promote its characteristics and the sale of the product. Retail stores and online shops can enable consumers to smell new products before purchase. Electronic encoding will also allow computer simulation and creation of new scents. Furthermore, existing scents can be edited to improve or enhance desirable characteristics. In the context of multimedia presentations, the electronic scent data may be embedded in a multimedia file to synchronize the production of scents along with visual and audio presentation of the multimedia content. For example, virtual reality games could now include scent for more realistic game play.

In another embodiment, the invention encompasses a method for representing a scent comprising the steps of profiling the scent based on one or more elements from an ordered set of elements to obtain a scent profile; and electronically representing the scent profile, wherein the scent may be substantially reproduced by an emitting device based on the electronically represented scent profile. The ordered set of elements can be linear in relationship, to permit substantially predictable reproduction of any scent by selecting the elements in selective quantity. The scent profile can be based on one or more elements from an ordered set of

elements that is independent of a mammalian olfactory system. Alternatively, the ordered set of elements is a set of scent components relative to a mammalian olfactory system.

The method of generating the scent profile can comprise the steps of analyzing the scent to determine one or more scent components that makes up the scent in reference to an olfactory space, and characterizing the scent based on the one or more scent components. The analyzing step of this method can include the step of determining the relative intensity of each scent component. The analyzing step of this method can also include the step of determining the one or more scent components based on selective binding of chemical components making up the scent to olfactory receptors in the olfactory space, and determining the relative intensity of interaction of the chemical components with the olfactory receptors, where the scent is characterized further based on the intensity of the interaction with the olfactory receptors. The scent components can be expressed by using at least one of the following type of components: receptor primary scent components, receptor complex scent components, receptor quasi-primary scent components, perceptive primary scent components and perceptive complex scent components.

The invention also encompasses electronic represention of a scent by creating or obtaining a vector representation of the scent profile. The vector representation can be based on a set of orthogonal basis vectors, each vector representing the status of the binding of an olfactory receptor in the olfactory space and the relative intensity of interaction thereof. The electronically represented scent profile can be electronically stored in a file for archival and/or transmission.

In another embodiment, the invention also encompasses a method for reproducing a scent comprising the steps of obtaining a electronic representation of the scent based on its scent profile, where the scent profile is based on one or more elements from an ordered set of elements; and activating one or more corresponding chemical components in accordance with the electronic representation of the scent profile, wherein the one or more chemical components when interacted with a user's olfactory receptors produce a perception of the scent to the user. The ordered set of elements can be linear in relationship, permitting substantially predictable reproduction of any scent by selecting the elements in selective quantity. The ordered set of elements can be a set of scent components relative to a mammalian olfactory system. The scent components can be defined based on selective binding of chemical components making up the scent to olfactory receptors. The electronic representation can comprise information representing the relative intensity of the chemical components to be activated, so as to result in interaction of the chemical components with olfactory receptors in

the appropriate relative intensity for a user to perceive the scent that is being reproduced. The one or more elements can be defined in relation to a set of scent components in an olfactory space, and the electronic representation can include information representing the relative intensity of the scent components in the scent profile. As in the previous embodiment, the electronic representation can a vector representation of the scent profile, and the vector representation can be based on a set of orthogonal basis vectors, where each of the vectors can represent selective binding to an olfactory receptor in the olfactory space and the relative intensity of interaction thereof to reproduce the scent. The scent components can be expressed as one or more of receptor primary scent components, receptor complex scent components, receptor quasi-primary scent components, perceptive primary scent components, and perceptive complex scent components. The electronic representation of the scent can be transmitted or received electronically via an information exchange network, such as the World Wide Web or the Internet.

The step of activating one or more corresponding chemical components in accordance with the electronic representation of the scent profile can be performed by a scent emitting device that operates on the electronic representation of the scent profile. The scent emitting device canoperates on one or more index values which correlate to the chemical components available at the emitting device.

As in the previous embodiment, the scent profile can based on elements from an ordered set of elements that is independent of a mammalian olfactory system. The ordered set of elements can be transformed into an ordered set of elements relative to a mammalian olfactory system.

In another embodiment, the invention encompasses a method for electronically creating a scent, comprising the steps of selecting two or more elements from an ordered set of elements that relates to an olfactory space, electronically manipulating the two or more elements to create a electronic representation of a scent that has the desired olfactory characteristic, and substantially reproducing the scent by an emitting device based on the electronic representation. The electronically manipulating step can comprise the step of combining the two or more elements in appropriate relative intensity to create the scent. The ordered set of elements can be linear in relationship, to permit substantially predictable scent characteristics by selecting the two or more elements in selective quantity. The ordered set of elements can be a set of scent components relative to a mammalian olfactory system, and the

two or more elements can defined within a set of scent components in an olfactory space. The scent components can be defined based on selective binding of chemical components making up the scent to olfactory receptors. The scent components can be represented electronically as a vector representation. The vector representation can be based on a set of orthogonal basis vectors, where each vectors represents selective binding to an olfactory receptor in the olfactory space and the relative intensity of interaction thereof, in order to reproduce the scent. The scent components can be expressed as one or more of receptor primary scent components, receptor complex scent components, receptor quasi-primary scent components, perceptive primary scent components, and perceptive complex scent components. The scent profile can be based on elements from an ordered set of elements that is independent of a mammalian olfactory system. The electronic manipulation step can include assigning one or more index values which correlate to scent emitting components available at the emitting device.

In another embodiment, the invention encompasses a method for editing or customizing a scent. This editing or customizing comprises the steps of electronically representing the scent based on one or more elements from an ordered set of elements; and performing one or more steps of adding, removing, or changing the relative intensity of one or more elements to obtain a modified scent.

In another embodiment, the invention provides for a method for identification of an object, a concept or an event, comprising the steps of providing a electronic scent file representing a scent, where the electronic scent file is based on one or more elements from an ordered set of elements and where the scent can be substantially reproduced by an emitting device based on the electronic scent file, and associating the scent to the object, concept or event, where the object, concept or event can be identified by reproducing the scent from the electronic scent file. The object, concept or event can be at least one of daily events in life, advertising, promotion, product, service, content delivery, hardware operating status, thought, location, physical object, tangible object and intangible object. "Daily events in life" can be, but are not limited to, experiences such as waking up, washing, walking down a street, entering a kitchen, entering a restaurant, entering a movie theater, and other such events.

In another embodiment, the invention provides a method for fingerprinting a scent. This method comprises the steps of representing the scent based on one or more elements from an ordered set of elements to obtain a scent representation, and electronically representing the scent representation to form a electronic fingerprint of the scent, where the scent can be substantially identified based on the electronic scent representation. The ordered set of elements can be linear in relationship. The scent representation can be based on an ordered set

of elements that is independent of a mammalian olfactory system, or the ordered set of elements can be a set of scent components defined relative to a mammalian olfactory system. The scent representation can include information concerning the relative intensity of each element. If the ordered set of elements is defined independently of a mammalian olfactory system, the ordered set of elements can be transformable into another ordered set of elements that corresponds to a mammalian olfactory system, such as an ordered set of scent components; the resulting profile can be in electronic form, able to be stored in a file for archival and/or transmission. The scent components can be expressed as one or more of receptor primary scent components, receptor complex scent components, receptor quasi-primary scent components, perceptive primary scent components, and perceptive complex scent components.

In another embodiment, the invention provides for a method for identifying the operating status of a hardware or software system, comprising the steps of providing a electronic scent file representing a scent, wherein the electronic scent file is based on one or more elements from an ordered set of elements and wherein the scent may be substantially reproduced by an emitting device based on the electronic scent file, and associating the scent to the operating status of the hardware or software system, whereby the operating status may be identified by reproducing the scent from the electronic scent file.

In another embodiment, the invention provides for a method for simulating events, comprising the steps of providing a electronic scent file representing a scent, where the electronic scent file is based on one or more elements from an ordered set of elements and wherein the scent may be substantially reproduced by an emitting device based on the electronic scent file, and associating the scent to an event, whereby the event is simulated by reproducing the scent from the electronic scent file. The event can be a daily event in life.

In another embodiment, the invention provides a system for transmitting a scent over an information exchange network, comprising means for profiling the scent based on one or more elements from an ordered set of elements to obtain a scent profile, means for electronically representing the scent profile, where the scent can be substantially reproduced by a scent emitting device based on the electronically represented scent profile, means for transmitting the electronically represented scent profile over the information exchange network, and means for receiving the electronically represented scent profile from the information exchange network. The information exchange network can be the Internet. The system can further comprise means for the identification of the scent from the electronically represented scent profile, and a scent emitting device structured and configured to operate on

the electronically represented scent profile which substantially reproduces the scent. The scent emitting device can comprise means to selectively vaporize one or more corresponding chemical components in an appropriate relative intensity in accordance with the electronically represented scent profile, where the one or more chemical components when interacted with a user's olfactory receptors produce a perception of the scent to the user. The scent emitting device is structured and configured to further operate on one or more index values which correlate to the chemical components that are made available at the emitting device.

The means for profiling the scent can comprise means for analyzing the scent to determine one or more elements that makes up the scent, and means for characterizing the scent based on the one or more elements. The means for transmitting can transmit the electronically represented scent profile in a standalone file or as part of another file and the electronically represented scent profile can be referenced in another file. The system can further comprising means for synchronizing the reproduction of the electronically represented scent profile to the execution of the another file. The electronically represented scent profile can be a electronic fingerprint of the scent.

The means for reproducing the scent can comprise means for transforming the scent profile to a scent profile that is based on an ordered set of elements relative to a mammalian olfactory system. The means for reproducing the scent can comprise a scent emitting device structured and configured to operate on the transformed scent profile to substantially reproduce the scent.

The invention also provides for an operating system for a scent emitting peripheral device, comprising means for processing a electronic representation of a scent profile, where the scent profile is created based on one or more elements from an ordered set of elements; and means for instructing the scent emitting peripheral device based on the electronic representation of the scent profile. This operating system can be a component of another operating system. The operating system can be in the form of a peripheral device driver. The ordered set of elements can a set of scent components relative to a mammalian olfactory system, and the elements can defined within a set of scent components in an olfactory space. The electronic representation can include information representing the relative intensity of the scent components in the scent profile. The scent components can be defined based on selective binding of chemical components making up the scent to olfactory receptors. The electronic representation can comprise information representing the relative intensity of the chemical components to be activated so as to result in interaction of the chemical components

with olfactory receptors in appropriate relative intensity for a user to perceive the scent that is being reproduced.

The operating system can use a electronic representation which is a vector representation of the scent profile. The vector representation can be based on a set of orthogonal basis vectors, where each vector represents selective binding to an olfactory receptor in the olfactory space and the relative intensity of interaction thereof, to reproduce the scent. The scent components can be expressed as one or more of receptor primary scent components, receptor complex scent components, receptor quasi-primary scent components, perceptive primary scent components, and perceptive complex scent components. The operating system can perform the step of activating by utilizing a scent emitting device that operates on the electronic representation of the scent profile, such as a scent profile based on elements from an ordered set of elements that is independent of a mammalian olfactory system. That set of elements can be transformed into an ordered set of elements relative to a mammalian olfactory system.

The invention further provides for a method of electronic distribution of scents, comprising the steps of providing an electronic depository of electronic representations of scents, where the electronic representations are electronic scent profiles each based on one or more elements from a ordered set of elements, authorizing user access to the depository, and providing access to the depository to authorized users. The depository can be made available for access via an information exchange network, such as the Internet or the World Wide Web.

The invention further provides for a method for delivering electronic content to a user. This method comprises providing the electronic content; referencing in the electronic content a electronic scent file representing a scent relating to the content, wherein the electronic scent file is a electronic representation of a scent profile that is based on one or more elements from an ordered set of elements; and offering the user an opportunity to activate a scent emitting device using the electronic scent file to generate the scent to provide a scent effect in addition to visual, audio, or tactile effects, thereby to improve user perception or enjoyment of the content.

The electronic scent file can made available to the user via a recorded medium. Alternatively, both the electronic content and the electronic scent file can be made available to the user online. The content can be presented using a electronic information file that refers to the electronic scent file; the electronic scent file can be functionally coupled to or included in the electronic information file. The electronic scent file can be applied to activate the scent emitting device upon user interaction of the electronic information file. The electronic

information file can contain a link to the electronic scent file, where the electronic scent file is applied when the link is activated by a user. Alternatively, the electronic scent file can activate the scent emitting device without user intervention when the content is displayed. The link can be to a server that serves the electronic scent file to the user upon the activation of the link. The server can include a depository of a plurality of electronic scent files, each corresponding to a different scent.

The electronic content can comprise video (visual) content, audio content, or both video and audio content, and the operation of the scent emitting device and the application of the electronic scent file can controlled such that the scent generated is synchronized to the events in the video content, audio content, or both the video and audio content during playback thereof. Examples of electronic content include, but are not limited to, movies, videos, Web pages, banner advertisements, music, books, reference works, maps, pictures and other images, and the state of a device or operating system. The electronic content can be provided in various formats, including, but not limited to, videotapes, audiotapes, cassette tapes, compact disks (CD's), digital video disks or digital versatile disks (DVD's), computer memory devices such as ROM, RAM, PROM, EPROM, hard disks, floppy disks, electronic files, software cartridges, Web pages, file servers, and the like.

The invention further provides a process for improving promotion of a product or service to a user, comprising the steps of providing a electronic scent file representing a scent relating to the subject matter of a promotion, and offering the user an opportunity to activate a scent emitting device by applying the electronic scent file to generate the scent to provide a sensory effect in addition to a visual and/or audio effect of the promotion. The electronic scent file can be a electronic representation of a scent profile based on one or more elements from an ordered set of elements. The actual product or service need not be made available to the user at the time of the promotion. The promotion can be presented online using a electronic promotion file that includes or refers to the electronic scent file, or where the electronic scent file is made available to the user upon user request. The promotion can be presented as an advertisement in a banner in a web browser interface, and the electronic scent file can be functionally coupled to the banner. Alternatively, the banner can be presented using a electronic banner file which includes the electronic scent file. The electronic scent file can activate the scent emitting device without user intervention when the banner is displayed., or the electronic scent file can activate the scent emitting device upon user selection of the banner. The banner can contain a link to the electronic scent file, so that the electronic scent file can be applied when the link is activated by a user. The link can be to a server that serves

the electronic scent file to the user upon activation of the link by the user. The server can include a depository of electronic scent files, each corresponding to a different scent.

The invention further provides for a process for facilitating user selection of a product or service, comprising the steps of identifying a scent associated with the product or service; providing a electronic scent file representing the scent; and offering the user an opportunity to activate a scent emitting device by applying the electronic scent file to generate a sampling of the scent. The actual product or service need not be made available to the user at the time of user selection of the product or service. The product or service can be offered online, with the electronic scent file made available to the user online to facilitate user selection of the product or service. The product or service can be presented using a electronic information file that refers to the electronic scent file. The electronic scent file can be functionally coupled to or included in the electronic information file

The electronic scent file can be applied to activate the scent emitting device either with or without user intervention when the electronic information file is displayed. The electronic information file can contain a link to the electronic scent file, wherein the electronic scent file is applied when the link is activated by a user, or the link can be to a server that serves the electronic scent file to the user upon activation of the link by the user. The server can include a depository of electronic scent files, each representing a different scent.

In another embodiment, the invention provides for an emitting device for generating a scent comprising means for receiving a electronically represented scent profile that is based on one or more elements from an ordered set of elements, wherein the scent profile represents the scent to be created; means for holding chemical components which may be selectively activated to create a perception of the scent; means for activating individual chemical components; and means for controlling the means for activating the individual chemical components in accordance with the electronically represented scent profile to generate the scent. The means for activating can comprise means for vaporizing the individual chemical components. The means for controlling can selectively vaporize the individual components in appropriate relative intensity in accordance with the scent profile. The emitting device can further comprise means for blending the vaporized chemical components, and can comprise means for delivering a stream of air to transport the vaporized chemical components to emit as the scent from the device. In one embodiment, the means for holding chemical components comprises interchangeable cartridge means for holding a limited set of chemical components required for generating scents relevant to a specific application. The means for controlling can comprise means for operating on one or more index values that correlate to the chemical

components that are made available at the emitting device. The scent profile can be based on elements from an ordered set of elements that is independent of a mammalian olfactory system, which can be transformed into an ordered set of elements relative to a mammalian olfactory system.

The emitting device can contain at least one of the chemical components in solid phase or in liquid phase. The liquid phase chemical component can pressurized for delivery for vaporization or as an aerosol.

In a further embodiment, the invention provides for a scent emitting device comprising a plurality of reservoirs for containing chemical components; one or more evaporation chambers in which chemical components evaporate and mix; a plurality of capillaries in fluid communication with the reservoirs and the evaporation chamber, where the chemical components flow through the capillaries to the evaporation chamber by capillary action; and control means for selectively regulating the flow of the chemical component through each capillary, so as to deliver the appropriate chemical components in the appropriate amount to the evaporation chamber to be evaporated and mixed to create a desired scent. The control means can comprise micro-valves, each regulating the flow through a capillary, and a controller structured and configured to selectively control the operation of the micro-valves based on input data that represents the scent to be created. The evaporation chambers comprises means for heating the chemical components delivered to the evaporation chamber. The scent emitting device can further comprise means for delivering a stream of air through the evaporation chamber to transport the evaporated chemical components to emit as the scent from the device.

In another embodiment, the invention provides for a scent emitting device comprising a plurality of reservoirs for containing chemical components; an evaporation platform on which chemical components are deposited, evaporated and mixed; a plurality of conduits in fluid communication with the reservoir and the evaporation platform, wherein the chemical components flow through the conduits to the evaporation platform; heating means for selectively heating the chemical component in each reservoir to facilitate delivery of the chemical component through the conduit and deposition on the evaporation platform; and control means for selectively regulating the flow of the chemical component through each conduit by selectively controlling heating of the heating means, so as to deliver the appropriate chemical components in the appropriate amount to the evaporation platform to be evaporated and mixed to create a desired scent. The scent emitting device can further comprise means for

heating the chemical components deposited on the evaporation platform. The chemical components can be deposited on respective regions on the evaporation platform and the means for heating the chemical components deposited can comprise means for selectively heating the regions to achieve the desired evaporation rate for the respective chemical components so as to achieve the desired intensity of the chemical components for mixing to create the scent. The control means can comprise means for controlling operations of the separate heating means in synchronization based on input data that represents the scent to be created.

In another embodiment, the scent emitting device further comprises means for delivering a stream of air over the evaporation platform, to transport the evaporated chemical components to be emitted as the scent from the device. Means can be included for moving the evaporation platform to transport the chemical components deposited thereon away from the conduits. The evaporation platform can be designed to transport the chemical components to a heating station downstream in the direction of travel of the evaporation platform, where the chemical components are selectively heated and evaporated at the heating station so as to minimize evaporation outside of the heating station. The evaporation platform can made of a fibrous sheet material or a high surface area material which absorbs the chemical components deposited thereon, such as blotting paper or gauze. The fibrous sheet material or high surface area material can in the form of a continuous roll, and the means for moving unwinds the roll at one end to let the material pass under the conduits and to the heating station, and winds the roll at another end to take up the material after it passed the heating station. Alternatively, the evaporation platform can comprise a plurality of pads, each positioned under a conduit, where each pad can be separately and selectively heated. Heating elements can include, but are not limited to, resistance heaters, streams of hot air or other gases, lasers, infrared emitters, lamps, and other devices capable of emitting, transmitting or imparting heat to a substance.

The control means of the scent emitting device can comprise means for controlling the operations of the heating means, the heating station and the means for moving the evaporation platform to synchronize the deposition of the appropriate chemical components in the appropriate amount and the evaporation thereof in the appropriate intensity, based on input data that represents the scent to be created.

In another embodiment, the invention provides for a scent emitting device which comprises a plurality of reservoirs for containing chemical components; an air flow tube operatively coupled to each reservoir; a plurality of conduits in fluid communication with the reservoirs and the air flow tube, in a manner wherein the chemical components are drawn through the conduits by air flow in the air flow tube; and control means for selectively

regulating the flow of the chemical component through each conduit, so as to deliver the appropriate chemical components in the appropriate amount to the air flow tube. The control means can comprise a plurality of valves, each regulating the flow through a conduit; and a controller structured and configured to selectively control the operation of the valves based on input data that represents the scent to be created. The device can further comprise means for delivering a stream of air through the air flow tube to transport the chemical components drawn from the conduit to emit as the scent from the device. The air flow tube can be a Venturi tube. The conduits can be comprised of tubes.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a schematic representation of the system for electronic recording, analysis, editing, and playback of scents;

Figure 2 is a schematic representation of an alternate embodiment of an emitter subsystem in which a deep well plate is used to store the scent reagents;

Figure 3A is a schematic representation of an alternate embodiment of an emitter subsystem in which a heating plate is used to vaporize scent chemicals;

Figure 3B is a schematic representation of an alternate embodiment of an emitter subsystem in which a resistor powered heating plate is used to vaporize scent chemicals with different regional temperatures;

Figure 4 is a schematic representation of an alternate embodiment of an emitter subsystem in which a blotter paper is used to vaporize scent chemicals over a stationary heating element;

Figure 5 is a schematic representation of an alternate embodiment of an emitter subsystem in which individual heated pads below each orifice is used to vaporize scent chemicals;

Figure 6 is a schematic representation of an alternate embodiment of an emitter subsystem in which a Venturi tube is used in conjunction with the fluid mechanical properties of moving air to vaporize scent chemicals;

Figure 7 is a schematic representation of an alternate embodiment of an emitter subsystem in which a high surface area medium, such as a gauze is used to vaporize scent chemicals; and

Figure 8A is a flow diagram depicting the process of representing a scent as a scent object.

Figure 8B is a flow diagram depicting the process of converting the vectors composing a scent object into a scent palette vector (left) or converting a scentographer's design effort into a scent palette vector (right).

Figure 9 is a class diagram for a scent object.

Figure 10 is an example of a scent object.

Figures 11A, 11B, and 11C are tabular representations of the example scent object.

Figure 12 is a diagram showing the chemical structure of gamma-terpinene.

Figure 13 is a pictorial representation of the sequence of scent digitization, broadcasting and synthesis.

Figure 14 is a pictorial diagram of some of the various applications of the electronic scent concept of the present invention.

Figure 15 is a histogram of computer-generated Euclidean differences for unlike scents.

Figure 16 is a histogram of computer-generated Euclidean differences for like scents.

Figure 17 is a histogram of computer-generated Euclidean differences for like scents.

Figure 18 is a histogram of computer-generated Euclidean differences for unlike scents.

Figure 19 is a histogram of computer-generated Euclidean differences for like scents for molecules with "strong" scents.

Figure 20 is a histogram of computer-generated Euclidean differences for unlike scents for molecules with "strong" scents.

Figure 21 is a histogram of computer-generated Euclidean differences for like scents using longer vectors for distance comparisons.

Figure 22 is a histogram of computer-generated Euclidean differences for unlike scents using longer vectors for distance comparisons.

Figure 23 is a histogram of computer-generated Euclidean differences for like scents using variations in the order in which entries were run.

Figure 24 is a histogram of computer-generated Euclidean differences for like scents using variations in the order in which entries were run.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present description is made for the purpose of illustrating the general principles of the invention and should not be construed to limit the invention.

The present system comprises various components. The system is structured and configured for enabling the following functions; the functions can be performed separately or in conjunction with each other:

1. capturing a physical sample of the scent;
2. analyzing the scent, by determining the interaction of the scent with olfactory receptors (e.g., quantitating the scent in terms of "olfactory space" vectors, by computer simulations of the interaction of the scent with olfactory receptors) by using a subjective scent analysis system, by using chemical analysis and/or computer modeling of the chemical components of the scent; or by using other techniques for scent analysis;
3. classifying, indexing, and/or electronically representing the scent and its components;
4. electronically transmitting and/or archiving the electronic representation of the scent;
5. reconstructing the scent from the electronic representation; and
6. emitting the scent.

Referring to FIG. 1, the system 10 is composed of two basic subsystems, (i) a recording subsystem 12 which comprises of a sensory subsystem 20 for capturing scents and an interpretation subsystem 30 for converting scents to digital or other electronic files and analyzing or editing those files electronically and (ii) an emission subsystem 40 for reproducing scents. The subsystems may be coupled electronically via an information exchange network, such as via a wide-area-network (WAN) 50 such as the Internet. Each of the subsystems are described below in detail.

Scent Analysis/Recording

One embodiment of the invention is directed to a system that detects, analyzes, records, interprets, transmits (in digital or other electronic representation) and re-creates scents. The scent analysis module of this system can be implemented in a variety of ways.

The detection of the presence of a scent is straightforward. In one embodiment, the user simply signals to the system (via a switch or other device) that a scent is present and

should be analyzed; the scent or scent-emitting object can be presented to the intake valve of the device (the "nose" of the system). Alternatively, the detection of the scent can be performed automatically, as it involves simply detection of a volatile compound not present in the "resting" state of the device. Thus, a baseline can be created by sampling the surrounding air, and analyzing the baseline using one or more of various chromatographic, spectroscopic, or other analytical techniques (including, but not limited to, gas chromatography, gas-liquid chromatography, mass spectroscopy, optical or UV spectroscopy, or ionizing radiation devices). Deviation from the baseline by a predetermined threshold signals that a scent is present and should be analyzed.

Analysis of the scent can be performed in a number of ways. Various embodiments of the scent analysis system are presented. Theories of how these embodiments operate are also presented, although it should be emphasized that the invention is not limited by any particular theory of olfactory perception or scent analysis. An "analysis module," as part of a system for scent analysis, is either one or more humans, or a device, which implements a method of representing a scent symbolically or electronically. Several such methods are described herein.

A "scent representation" is defined as any way of representing a scent or describing a scent in a manner which can be recorded, stored or transmitted, including, but not limited to, representations in terms of receptor primary scent components, receptor quasi-primary scent components, receptor complex scent components, perceptive primary scent components, perceptive quasi-primary scent components, perceptive complex scent components, scent profiles, scent fingerprinting, a list of one or more odorant compounds which may include relative proportions and/or concentrations of the one or more compounds, or any combination of the foregoing representations, as defined herein. A "scent object" is an object which enables storage and/or transmission of a scent representation electronically, such as on a computer or a computer-readable medium, and is described herein in further detail.

An "odorant substance" or an "odorant" is defined as any substance which gives rise to a perceptible odor. An odorant substance may be a pure chemical compound, or it may be a mixture of chemical compounds. A "user" refers to a subject, preferably a mammalian subject, more preferably a human subject, and "sampling" of an odorant by a user simply refers to smelling the scent, that is, perceiving the olfactory stimulus of the scent.

Olfactory Space

The sensory subsystem comprises a series of olfactory receptors, which selectively bind with the chemical component(s) making up a scent. A scent can be characterized in terms of which of the approximately 1,000 or more olfactory receptors the scent component(s) bind to, and the strength of the interaction of the component(s) with those receptors. Each olfactory receptor can be considered an orthogonal basis vector; the entire set of olfactory receptors can be considered a set of basis vectors spanning "olfactory space." This is analogous to vectors pointing along the x, y, and z directions in three-dimensional space, where any point in space can be represented by a combination of the x, y, and z basis vectors (with each of the x, y, and z vectors multiplied by the appropriate scalar quantity). The intensity of interaction of a scent with an olfactory receptor determines the magnitude of the vector along that particular "axis" in olfactory space. Thus, every scent can be uniquely described by a vector representation in olfactory space.

A representation of a scent in such a manner that the scent can later be re-created is defined as scent profiling. The aforementioned vector representation is one example of a scent profile.

An olfactory space can also be defined in terms of odorant substances. That is, given one or more odorant substances, those substances span an olfactory space which encompasses the entire range of scents that can be created from any combination in any intensity of those substances. Thus an olfactory space can be defined in terms of the response of a sensor, such as an olfactory receptor, or in terms of odorant substances available to create scents.

Primary Scents

For the purposes of this invention, a receptor primary scent component is defined as a chemical that activates one and only one scent receptor. A receptor complex scent component is defined as a chemical that activates more than one scent receptor; the receptor complex scent component can activate each of the scent receptors to different degrees, to equal degrees, or can activate some receptors to the same degree and others to different degrees.

"Activating a scent receptor" is defined as the ability of a chemical to interact with a given olfactory receptor in such a manner as to communicate to a cell expressing such a receptor that the chemical is present external to the cell. Olfactory receptors are proteins which fall in the class of seven transmembrane domain G protein-coupled receptors, and are found in olfactory neurons *in vivo*. Binding of an odorant to an olfactory receptor causes second messenger systems to become activated in the cell, leading to increased cellular

production of second messenger molecules such as cyclic AMP. These second messenger systems in turn lead to the depolarization of the olfactory neuron, or other changes in the state of the neuron, which provides the signal to the nervous system that the odorant has been detected.

With a complete set of receptor primary scent components, any scent can be recreated with the knowledge to the degree to which it activates each olfactory receptor. The instant invention encompasses such complete sets of receptor primary scent components. Other embodiments of the invention encompass sets of receptor primary scent component chemicals which provide the ability to re-create a particularly desired subset of scents, but not necessarily all possible scents. Such sets are "incomplete" sets of receptor primary scent components, but retain significant utility in that a particularly desired subset of scents, for example, the subset of perfume scents, or the subset of wine scents, or the subset of scents which are sensed in a forest, can be reproduced. In terms of "olfactory space," such an incomplete set spans an "olfactory subspace," that is, only a portion of olfactory space. Still more embodiments encompass sets of receptor primary scent component chemicals which provide the ability to approximate particular scents, while not necessarily exactly re-creating the activation profile of the particular scents. These sets are also "incomplete" sets of receptor primary scent components, and differ from the previous description of incomplete sets in that they approximate, rather than re-create, the receptor activation profile of the desired subset of scents.

In some cases, a receptor complex scent will be an acceptable approximation to a receptor primary scent. That is, if a given receptor complex scent activates a first scent receptor strongly, but activates other scent receptors less strongly, it can be considered an approximation to a receptor primary scent component for the first receptor. Such a receptor complex scent component is described by the term receptor quasi-primary scent component. One embodiment of the invention encompasses sets of receptor quasi-primary scent component chemicals suitable for re-creating all scents. Another embodiment of the invention encompasses sets of receptor quasi-primary scent component chemicals suitable for re-creating a particularly desired subset of scents, but not necessarily all possible scents. Yet another embodiment encompasses sets of receptor quasi-primary scent component chemicals which provide the ability to approximate particular scents, while not necessarily exactly re-creating the activation profile of the particular scents.

The identification of receptor primary or quasi-primary scent component chemicals provides the most conceptually straightforward method of re-creating scents. However,

another embodiment of the invention encompasses the use of receptor complex scent components for re-creating scents. An example of such an embodiment would be re-creation of a scent that activates olfactory receptors designated OR1, OR2, OR3, OR4, OR5 and OR6 (for the sake of illustration, it is assumed that the olfactory receptors are stimulated to an equal extent). If one is in possession of two receptor complex scent component chemicals (RCSC's) where RCSC1 activates OR1 and OR5, and RCSC2 activates OR2, OR3, OR4, and OR6, then one can reproduce the original scent by mixing RCSC1 and RCSC2 to re-create the original olfactory receptor activation profile. In practice, the profiles of various receptor complex scent components will be much more complicated than the forgoing example, and components which inhibit olfactory activation as well as stimulate activation can be included in the sets. However, once receptor activation profiles of sufficient receptor complex scent components are known, computer algorithms can be utilized to create the appropriate combination of receptor complex scent components. Using vector representations of the olfactory receptor activation profiles for a set of receptor complex scent components, one can create linear combinations of such receptor complex scent components in order to represent a particular scent. For the example given above, such a vector representation would look like (1, 0, 0, 0, 1, 0) for the first receptor complex scent component and (0, 1, 1, 1, 0, 1) for the second receptor complex scent component, while the vector representation of the scent to be re-created is (1, 1, 1, 1, 1, 1). If x_1 and x_2 are the relative proportions of the first receptor complex scent component and the second receptor complex scent component, respectively, to be combined to re-create the scent, then the problem can be represented as a series of linear equations:

$$
\begin{aligned}
1x_1 + 0x_2 &= 1 \\
0x_1 + 1x_2 &= 1 \\
0x_1 + 1x_2 &= 1 \\
0x_1 + 1x_2 &= 1 \\
1x_1 + 0x_2 &= 1 \\
0x_1 + 1x_2 &= 1
\end{aligned}
$$

and the solutions for x_1 and x_2 are $x_1 = 1$, $x_2 = 1$. Solutions to systems of linear equations have been thoroughly studied and many algorithms are available for implementation on computers, including algorithms which evaluate the accuracy of an approximate solution when an exact solution cannot be determined. (See, e.g., Dettman, J.W., *Introduction to Linear Algebra and Differential Equations*, Dover Pubs., 1986; Press W.H. et al., *Numerical Recipes in C: The*

Art of Scientific Computing, 2nd ed., Cambridge University Press, 1993; Vetterling (ed.) *Numerical Recipes in C: The Art of Scientific Computing/Disk V 2.02*, Cambridge University Press, 1997.) These methods can also be used to determine whether a set of receptor complex scent components is suitable for re-creating a given scent. For example, if the scent to be recreated is represented by the vector (1, 1, 1, 1, 1, 2), there will be no solution to the resulting system of linear equations using the two receptor complex scent components in the illustration above. In this instance, one or more additional receptor scent components will need to be identified in order to be able to re-create the scent in terms of the receptor primary scent components. Alternatively, the scent represented by (1, 1, 1, 1, 1, 1) may be an acceptable approximation to the scent represented by (1, 1, 1, 1, 1, 2). Integers are used in this example for clarity, but the vectors can contain any real number representing a measured intensity; for example, (1.1, 0.997, 1.08, 1.2, 0.88888..., 2.00001) may be an acceptable approximation to the scent represented by (1, 1, 1, 1, 1, 2).

It will be readily appreciated that the choice of a complete set of receptor primary, quasi-primary, or complex scent component chemicals (capable of generating all scents) versus a partial set of receptor primary, quasi-primary, or complex scent component chemicals (capable of generating, exactly or approximately, a subset of scents) depends on the application for which scent re-creation is desired.

A special category of receptor scent components are chemicals which bind to a receptor without activating it. If these non-activating chemicals prevent chemicals which do activate the receptors from binding, the non-activating chemicals act to "turn off" those receptors. These non-activating chemicals, or receptor binding antagonists, are particularly useful in editing scents, as they can be added to a scent to attenuate or eliminate particular aspects of the scent. In the vector example above, if a particular receptor antagonist blocks OR2, OR3, and OR4, but not OR1, OR5 or OR6, then it can be represented in vector format as (0, -1, -1, -1, 0, 0). In the reproduction of (1, 1, 1, 1, 1, 2) from the vectors (1, 0, 0, 0, 1, 0) and (0, 1, 1, 1, 0, 1), the following combination can be used: 1 x (1, 0, 0, 0, 1, 0) + 2 x (0, 1, 1, 1, 0, 1) + 1 x (0, -1, -1, -1, 0, 0) to yield the vector (1, 1, 1, 1, 1, 2). In some instances, enough of a particular receptor binding antagonist is used to eliminate any possibility of activation by a receptor scent component, in which case the vector entry for the receptor(s) which are blocked by that antagonist contains 0 in the vector position corresponding to that receptor(s).

Perceptive primary scents are defined as scents that give a single scent perception, for example, the scent "lemon" as perceived by a human. A perceptive primary scent can be

composed of one or more receptor primary scent components, one or more receptor complex scent components, or a mixture of one or more receptor primary scent components and one or more receptor complex scent components. Since perceptive primary scents are to some extent subjective, identification of perceptive primary scents can be performed by using a panel of subjects who evaluate and describe scents. A perceptive complex scent is made up of more than one perceptive primary scent. The boundaries between a perceptive primary scent and a perceptive complex scent are also to some extent subjective; for example, one person may describe a scent as "pizza," while another person may describe the same scent as "sausage, cheese and tomato sauce." That is, one person may perceive a scent as a perceptive primary scent for "pizza," while another person may perceive the same scent as a perceptive complex scent made up of several individual perceptive primary scents. Perceptive quasi-primary scents can be defined as scents that give one main scent perception, but other minor scent impressions; for example, a chocolate donut may give the main scent perception of "chocolate" and a minor scent impression of "fried dough." In order to standardize perceptive scents,at least one person, or a panel of five or more, preferably ten or more, more preferably fifty or more, still more preferably one hundred or more, people can be surveyed to label various perceptive scents. When a plurality, preferably a majority, more preferably 66 2/3 % or greater, still more preferably 95 % or greater, even more preferably 99% or greater, of a panel identifies a scent as the same scent (e.g., of a panel of 100 people, 95 describe a scent as "pizza," while the other 5 describe the scent otherwise), the scent can be labeled as a perceptive scent (the perceptive scent can be primary or complex, depending on whether the panel identifies it as a single scent or a mixture of scents).

It should be noted that any group of one or more odorants can function as perceptive scent components (whether primary, quasi-primary, or complex), and thus any one or more group of odorants defines an olfactory space. Each individual odorant can independently be either a pure chemical, or a mixture of chemicals. The olfactory space may be extremely small; for example, if the group of odorants consists of one chemical, e.g. acetic acid, the olfactory space defined by the group will be one-dimensional. The perceptive scent may be described as "vinegary", and the strength of the scent may be rated according to how much acetic acid is perceptible to the person(s) perceiving the scent. In preferred embodiments, however, many perceptive primary, quasi-primary, and complex scent components will be used. An example of a set of perceptive scent components is given in Table 1. In that table, allyl methyl trisulfide is assigned the single perceptive scent label "garlic," and thus can be considered a perceptive primary scent component, while 2,6-dimethyl-3-ethylpyrazine has a

variety of scent descriptors associated with it, and can be considered a perceptive complex scent component.

In fields where existing classification schemes already exist, the perceptive primary and complex scents can be indexed according to those schemes. For example, the SFP (Société Française des Parfumeurs) has drawn up a classification system based on 5 main groups, sub-divided into classes. A wine aroma wheel has been developed for describing the aroma of wines; see A.C. Noble, R.A. Arnold, J. Buechsenstein, E. J. Leach, J.O. Schmidy, and P.M. Stern, "Modification of a standardized system of wine aroma terminology", *American Journal of Enology and Viticulture,* 38/2 (1987). Such classifications can be used for selecting perceptive primary scents and used as guides for combining the scents.

These methods for representing scents rely on a set of ordered elements, such as a list of receptors which can be activated by a scent, or a list of perceptive primary scents, or an array of individual detectors, in order to describe the scent. In one embodiment, the invention embraces profiling the scent (or equivalently, creating a scent profile) based on one or more elements from such an ordered set of elements. These scent profiles can then be conveniently converted to electronic form to enable the scent profile to be electronically represented, and to enable the scent to be re-created or approximated by an emitting device. The profiling of the scent can be performed analyzing the scent in terms of scent components and then describing the scent in terms of those scent components; e.g., a description of the scent in terms of an ordered set of elements consisting of olfactory receptors would provide a vector representation in terms of receptor primary or quasi-primary scent components, such as that illustrated herein, which would then characterize the scent in terms of the selective binding of the scent to the olfactory receptors, more preferably in terms of the relative intensity of selective binding of the scent to the olfactory receptors. Preferably the scent components define an olfactory space or an olfactory subspace, where any scent, or any member of a subset of scents, can be represented by combinations of the scent components. The scent components can be receptor primary scent components, receptor quasi-primary scent components, receptor complex scent components, perceptive primary scent components, perceptive quasi-primary scent components, perceptive complex scent components, or any combination thereof. Preferably, vector representations of the scents are expressed in terms of orthogonal basis vectors which define an olfactory space; where each basis vector represents the binding of a scent to an olfactory receptor. The scent so represented can be stored electronically in a file for scent archival, transmission, and manipulations.

In one embodiment, the elements of the ordered set are linear in relationship, as in the matrix illustration above. The members of the ordered set of elements can be related to a mammalian olfactory system; for example, the members of the ordered set can represent basis vectors in olfactory space, where each member of the set describes the intensity of interaction of a scent with an olfactory receptor. Alternatively, the members of the ordered set of elements can be independent of a mammalian olfactory system; the ordered set of elements can represent the intensity of interaction of a scent with artificial detectors, such as the "artificial nose," or artificial scent-sensing device, described in U.S. Patent Nos. 5,571,401, 5,698,089, 5,788,833, 5,891,398 and 5,911,872. The artificial nose described therein is comprised of several polymeric scent-sensing elements (response elements), each with different response profiles to various volatile compounds. The response elements of the scent-sensing device thus define and span an olfactory space.

Scent Objects

As previously defined, a "scent object" is an object which enables storage and/or transmission of a scent representation electronically, such as on a computer or a computer-readable medium. Scent objects provide a particularly useful manner in which scent profiles can be represented on a computer in a form suitable for analysis, editing and re-creation of the scent. A scent object can be implemented in widely differing embodiments, and the examples of scent objects and implementations of scent objects given herein are not intended to limit the definition of scent objects. In one embodiment, a scent object can be implemented in electronic form as a data structure on a computer.

Various components of one example of a data structure implementation of a scent object are now described with reference to the Figures. It should be noted that the scent object can comprise any one or more of the components described in the following section (the Hedonic Vector, Polymer Vector, Screening Vector, Chemical Structure Similarity Vector, Docking Vector, or Scent Palette Vector).

Fig. 8A is a flow diagram depicting the process of representing a scent in a manner suitable for re-creation of the scent. A scent-emitting source can be sampled by a panel of human subjects, that is, by human sensory evaluation, and the result of such evaluation expressed as a "Hedonic Vector." A headspace sample of a scent can be fed to an artificial sensor device (such as the artificial scent-sensing device ("artificial nose") described in U.S. Patent Nos. 5,571,401, 5,698,089, 5,788,833, 5,891,398 and 5,911,872) for analysis of its interaction with that device; the result of this evaluation and detection is expressed as a

"Polymer Vector." The headspace sample can also be screened by receptor-ligand interaction using a panel of olfactory receptors as described above; the result is expressed as a "Screening Vector." Finally, the headspace sample can be broken down by gas chromatography (or other separation techniques well-known in the art of analytical chemistry) as a scent object. The Hedonic, Polymer, Screening, Chemical Structure Similarity and Docking vectors are described in more detail below.

The Scent Palette vector represents the fragrant chemicals available for output. As shown in Fig. 8B, by using a transformation algorithm, the representation of the scent object as a Hedonic, Structure, Screening, Polymer or Docking vector is transformed into a Scent Palette vector. The Scent Palette Vector indicates which of the available fragrant chemicals at the emitter should be mixed together, as well as the appropriate amounts of each fragrant chemical to use. Alternatively, a human operator or computer program may seek to design a scent using the fragrant chemicals available at the emitter. By combining available scents in the Scent Palette, using scent mixer software, a scentographer can design custom scents.

Fig. 9 is a class diagram for a scent object indexed using one or more of the Hedonic, Structure, Screening, Polymer or Docking vectors. Fig. 10 is an example of a scent object for a particular scent, the scent of the chemical gamma-terpinene. As illustrated in Fig. 9, the top level field of the scent object, listed as "Scent," contains basic identification, such as the name of the scent, the REMINISCENTS™ ID number for the scent (a unique number assigned to the scent; REMINISCENTS™is a trademark of DigiScents, Inc., Oakland, CA, for a scent name, scent label, or scent descriptor), alternate names which may be assigned to the scent, a description of the scent, and a sensory index value. The example of Fig. 10 lists the name of the scent as "Lemony;" the REMINISCENTS™ ID number for the scent as RID-7642; alternate names for the scent as "citrusy, fruity, sweet;" the description of the scent as "scent of lemon," and the sensory index as 54. The sensory index describes the scent in relation to a particular scheme of scent classification, such as the aroma wheel described in A.C. Noble, R.A. Arnold, J. Buechsenstein, E. J. Leach, J.O. Schmidy, and P.M. Stern, "Modification of a standardized system of wine aroma terminology", *American Journal of Enology and Viticulture*, 38/2 (1987). In the discussion below of the scent object diagrams of Figs. 9 and 10, at times the scent described by the scent object is called the "sample scent" when necessaryto distinguish it from other scents used in describing the components of the scent object.

The Hedonic Vector, which classifies the scent by using human olfactory perception, implements an embodiment of the invention using a set of odorant descriptors--e.g., "lemony," "citrusy," "vinegary," etc.--to generate a scent profile. Odorant descriptors are defined as any descriptive term suitable for describing a scent, such as common terms for scents, specialized terms used in various scent arts for particular scents, or terms for particular notes of a scent. Alternatively, since the odorant descriptors are associated with odorant substances of a particular reference set, the Hedonic Vector is also an implementation of an embodiment of the invention where a set of odorant substances sampled by one or more humans is used to create the scent profile. The Hedonic Vector contains two values, as illustrated in Fig. 9: a Reference Set ID, and a Preferred tag. A Reference Set is a collection of scents against which a sample scent is described. Many different Reference Sets can be compiled, and the Reference Set ID indicates which Reference Set is being used for that particular Hedonic Vector. As indicated in Fig. 9, a single sample scent can have n Hedonic Vectors associated with it, where n is an integer greater than or equal to one if Hedonic Vectors are being used to represent the sample scent. (Clearly, a value of zero for n would indicate that Hedonic Vectors are not being used to represent the scent.) The Preferred tag indicates whether that Reference Set is the preferred set for description of the particular sample scent being described. As depicted in Fig. 10, Hedonic Vector 1 refers to the Reference Set HVID 0921, and the Preferred tag is true, indicating that Reference Set HVID-0921 is the preferred set for describing the "lemony" scent; Hedonic Vector 2 refers to the Reference Set HVID-0498, and the Preferred tag is false, indicating that Reference Set HVID-0498 is not the preferred set for describing the "lemony" scent. An example of the generation of a Hedonic Vector is given in Example 2.

As previously described, the Reference Sets are collections of scents against which a sample scent is described. A single Reference Set can be a collection of one or more scents, two or more scents, five or more scents, or ten, twenty, twenty-five, thirty, forty, fifty, or seventy-five or more scents, or one hundred or more scents. As indicated in Fig. 9, a single Hedonic Vector can have n Hedonic Values, where n is the number of scents composing the Reference Set. A panel of human subjects is exposed to both the sample scent and each scent contained in the Reference Set. The panel rates the sample scent against each scent contained in the Reference Set using a numeric scale. The individual scents which compose the Reference Set are given reference numbers for identification, called the Hedonic Category ID. In one implementation of this rating, the human subject panel rates the sample scent against each scent in the Reference Set on a scale from 1 to 100, with 1 being "very similar/identical"

and 100 being "very dissimilar." The various ratings from each subject are then averaged together to determine the Hedonic Score for that particular scent in the Reference Set. The ratings can be adjusted as desired, for example, by dividing by 100 to scale the numbers between 0 and 1. Alternate numbers and evaluations can be used, for example, -100 for most unpleasant to +100 for most pleasant.

In Fig. 10, two examples of Hedonic Vectors and their corresponding Hedonic Values are shown. For Hedonic Vector 1, Hedonic Values 1 are given. When the "lemony" sample scent (gamma-terpinene) is compared against the "tart" scent in the Reference Set, it is rated 65.02 by the human subject panel. When the "lemony" sample scent is compared against the "Strong" scent in the Reference Set, it is rated 33.11 by the human subject panel. These values make up the Hedonic Values 1, which in turn is part of the Hedonic Vector 1.

The scents in the Reference Set are chosen to be representative of a particular scent property; for example, a solution of 5% acetic acid in water can be chosen to correspond to the reference scent "vinegary." A particular Reference Set may often be more appropriate for comparison to the sample scent than other Reference Sets, which is indicated by the Preferred tag in the Hedonic Vector.

The Structural Vector is illustrated in Fig. 9, and an example of the Structural Vector is illustrated in Fig. 10 for gamma-terpinene. The top level Scent field references a Compound field. In the example for gamma-terpinene, there is only one chemical compound present in the scent; however, for scents made up of multiple chemical components, there can be as many Compound fields as there are identifiable chemical components. In circumstances where only a few compounds are necessary to reproduce or approximate a scent, there can be fewer Compound fields than the number of identifiable chemical compounds in the scent, while if a scent is altered or enhanced by adding components not normally present in the scent, there can be more Compound fields than the number of identifiable chemical compounds in the original scent. These latter examples, however, are best described as "approximate Scent Objects" and "altered (or edited, or enhanced) Scent Objects" to distinguish them from the Scent Object corresponding to the original sample scent.

The Compound field is composed of several values. Fig. 9 illustrates a Compound field composed of: a chemical ID number, which can be arbitrarily assigned, or alternatively, can be a standardized number such as the Chemical Abstracts Registry Number for the compound; a SMILES (Simplified Molecular Input Line Entry Specification, see Weininger, D., "SMILES 1. Introduction and Encoding Rules", *J. Chem. Inf. Comput. Sci.* (1988) 28, 31) representation of the chemical; the Vapor Pressure of the chemical at a particular Vapor

Temperature; the Flash Point of the chemical; the Viscosity Rating and Oxidation Rating of the chemical on a predetermined or standardized scale; and the Solubility of the chemical. Other properties can be added to the Compound field. Fig. 10 illustrates a Compound field with the particular values for gamma-terpinene included.

As the Compound field contains the chemical-based information regarding the sample scent, the Structural, Receptor Docking, Receptor Screening, and Polymer Reactivity Vectors can be associated with the Compound field. It should be noted that since the Polymer Reactivity Vector and the Receptor Screening Vector can be recorded either for each individual compound in the scent, or for the scent as a whole, that those vectors can be associated with either the Compound field or the top level Scent field, or with both fields. The Structural and Receptor Docking Vectors are dependent on the chemical structure of each component of the sample scent, however, and must be associated with the Compound field.

The Polymer Reactivity Vector is an implementation of an embodiment of the invention where a set of response elements in an artificial scent-sensing device (an "artificial nose") is used to generate a scent profile. The Polymer Reactivity Vector is a vector which describes interaction of the chemical contained in the Compound field with an artificial scent-sensing or "artificial nose" type of detector, such as those described in U.S. Patent Nos. 5,571,401, 5,698,089, 5,788,833, 5,891,398 and 5,911,872. The artificial scent-sensing devices/artificial noses of those references detect chemicals by an array of polymeric materials, which respond to the presence of the chemical in a detectable manner. The Reference Set ID of the Polymer Reactivity Vector identifies the particular device which is used to detect the chemical compound, while the Preferred tag indicates whether that device is the preferred detector for that chemical. The Polymer Reactivity Value records the response/reaction of a single polymeric element of the artificial scent-sensing ("artificial nose") array; the element is identified by a Reference Polymer ID, and the quantitative response is recorded in the Reactivity Score. When an artificial scent-sensing device ("artificial nose") is used having non-polymeric response elements or detectors in addition to or instead of polymeric response elements, the Polymer Reactivity Vector and Polymer Reactivity Value are easily generalized to such non-polymeric response elements or detectors.

The Receptor Screening Vector, which can be associated with the Scent field, the Compound field, or both, as illustrated in Fig. 9, refers to Reference Sets of olfactory receptors used for screening chemical compounds for interaction with the olfactory receptors comprising the set. The Receptor Screening Vector is an implementation of an embodiment of the

invention which uses sets of olfactory receptors, preferably mammalian olfactory receptors, more preferably human olfactory receptors, to generate a scent profile. The Reference Sets are identified by Reference Set ID numbers, and the Preferred tag indicates whether that Reference Set of olfactory receptors is the preferred Reference Set for that particular sample scent, as each Scent or Compound may be described by several Reference Sets. The Receptor Screening Value indicates the intensity of interaction with the various receptors making up the particular Reference Set used. Each individual receptor is identified by a Reference Receptor ID number, with the screening score indicating the intensity of interaction of the compound with that receptor.

The Receptor Docking Vector, which is associated with the Compound field, describes the Reference Set of receptor molecular structures against which an individual chemical compound is described. The Receptor Docking Vector is an implementation of an embodiment of the invention which uses sets of olfactory receptor structures, modeled or simulated on a computer, to generate the scent profile. Since the chemical structure of the compound to be docked against the set of modeled olfactory receptor structures must be known to use the docking software, the Receptor Docking Vector must be associated with a specific chemical compound. The Receptor Docking Vector contains a Reference Set ID and a Preferred tag, to indicate whether the set of receptors identified by the Reference Set ID is the preferred set to use to describe the particular chemical. Each Receptor Docking Vector refers to one or more Reference Docking Values; the Values are composed of a Reference Receptor ID (which indicates the specific receptor into which the chemical is being docked) and the Docking Score (which indicates the interaction of the chemical with the receptor, as determined by the docking program used; these scores can be reported as individual real numbers (scalars) or may be reported as vectors, vectors of vectors, or other mathematical representations of the output of the docking program). Docking methods and software packages are described in more detail herein.

The Structural Vector represents a comparison of the chemical described by the Compound field with reference compounds. The Structural Vector is an implementation of an embodiment of the invention where a set of odorant substance structures modeled on a computer is used to generate the scent profile. The Structural Vector contains the Reference Set ID, which indicates which set of reference compounds is being used for comparison, and the Preferred tag, which indicates whether that set is the preferred set for description of the particular chemical of the Compound field. For each Reference Set, a Chemical Similarity Value is determined for the particular chemical of the Compound field as compared to the

chemicals comprising the Reference Set, and a Similarity Score is assigned. As with the Docking Score in the Reference Docking Value, the Similarity Score may be reported as individual real numbers (scalars) or may be reported as vectors, vectors of vectors, or other mathematical representations of the output of the chemical similarity program. Methods for assessing chemical similarity, and computer programs suitable for such methods, are described in more detail herein; see Example 3 for a description of a procedure for comparing the similarity of scent molecules.

The Palette Vector of the scent object differs from the other vectors in that it represents the scent in terms of the chemicals available for re-creation of scents, rather than by some characteristic, measurement or analysis of the scent itself. The Palette Vector is an implementation of an embodiment of the invention which uses a set of odorant substances available in a scent emitting unit to generate the scent profile. The other vectors can be transformed into Palette Vectors by algorithms; alternatively, a human operator (a scentographer) may design a scent using the available chemicals, much as an artist paints a picture from the colors available on the artist's palette. The Palette Vector refers to a Reference Set ID, which identifies a set of chemicals available for scent re-creation, and has a Preferred tag, which indicates whether that particular set of chemicals is the preferred set for re-creation of that scent. For each member of the reference set, Palette Values are provided which have a Reference Well ID value, indicating which well of a multi-well plate contains the chemical to be added to the overall mixture, and a Quantity value, indicating how much of that chemical to add to the overall mixture. When emitters utilizing storage devices other than a multi-well plate are used, the Reference Well ID can simply refer to a physical location on a cartridge, plate, or other device where the desired scent is located.

In one variation of this embodiment of a scent object, the scent object is composed only of the Palette Vector and the appropriate identifying information in the Scent field of the data structure. The odorants available in the scent emitter comprise an ordered set of elements, and the scent to emit is described by the Palette Vector in terms of that ordered set of elements. Thus, the Palette Vector (and associated identifying information) can serve as a complete scent representation in this embodiment.

The various representations of the scent profile can be readily transformed or interconverted. A scent profile generated by one method can be generated by another method, and a transformation matrix derived which will convert one representation to another. The most useful transformation is between a scent profile based on a set of mammalian olfactory receptor structures modeled on a computer, a set of odorant substances sampled by one or

more humans, a set of odorant substance structures modeled on a computer, a set of odorant descriptors, or a set of response elements in an artificial scent-sensing device, into a scent profile based on a set of odorant substances available in the emitting device. As an example, the odorant substance making up a Palette Vector in a scent object can be tested using an artificial scent-sensing device or artificial nose, to determine which sensing elements are activated by each odorant substance used for emission. When a sample scent is presented to the artificial nose, the activation pattern it produces in the scent-sensing elements is recorded, and a computer program is used to produce the closest approximation to that scent using the odorant substances available for emission (for which the response of the artificial nose is already known). Other transformations--e.g., from the artificial nose's set of response elements to the pattern of activation of a set of mammalian olfactory receptors--are also possible, and can have important uses; for example, a transformation from a scent profile generated using a set of odorant substance structures modeled on a computer into a scent profile based on odorant descriptors can identify scents which have negative scent descriptors.

The Appendix contains code (in C and Java) for computer implementation of a scent object.

Figs. 11A, 11B, and 11C provide tabular representations of the scent objects.

Selecting Chemicals for Scent Re-creation

A scent which has been represented as a set of basis vectors in olfactory space can in principle be re-created simply by mixing or synthesizing the receptor primary scent components, receptor quasi-primary scent components, or receptor complex scent components needed to activate the olfactory receptors in the same pattern as the original scent. Such an approach requires 1) a method to generate a representation of the original scent in olfactory space, and 2) suitable receptor primary scent component chemicals which can be mixed in the appropriate manner.

A scent which has been represented as a mixture of perceptive primary scents can be re-created by mixing or synthesizing the receptor primary scents needed to re-create the perceptive complex scent. Such an approach requires 1) a method to generate a representation of the original perceptive complex scent in terms of perceptive primary scent components, and 2) suitable perceptive primary scent component chemicals which can be mixed in the appropriate manner.

Identification of receptor scent components can be performed by various methods. One such method assays the interaction of candidate components with each olfactory receptor.

The receptors can be expressed *in vitro* and assays can be set up to monitor the interaction of various candidate components with each individual receptor. Chemicals which activate one and only one olfactory receptor are receptor primary scent components, while chemicals which activate more than one olfactory receptor are receptor complex scent components (and can possibly be receptor quasi-primary scent components, depending on the activation profile it displays with the olfactory receptors). Such an approach can use methods known in the art, for example those of Breer *et al,* Ann. N. Y. Acad. Sci. (1998) 855:175-81 or Malnic *et al., Cell* (1999) 96(5):713-23. Breer *et al.* expressed olfactory receptors in Sf9 cells and evaluated the second-messenger response to various odorants. Malnic *et al.* isolated olfactory neurons from mice and utilized calcium imaging to study the response of the neurons to different odorants, while using RT-PCR to determine which olfactory receptor was expressed in the neuron under study. U.S. Patent No. 5,798,275 describes a method for evaluating interaction of compounds with members of a reference panel of proteins.

Alternative methods for representing the scent can also be employed. As described above in the context of Scent Objects, a sample scent can be compared to a Reference Set of scents to create a Hedonic Vector representation of the sample scent. The particular scents in the Reference Set, along with their Hedonic Values, can then serve as a blueprint for re-creation of the sample scent. In one embodiment, the Reference Set scents simply serve as the Palette Vector set of scents, to be combined in the amounts indicated by the Hedonic Values. When a Structural Vector is used to represent a compound, the particular structures used for the Reference Set, along with their Chemical Similarity Values, can be used in an analogous fashion.

Once the electronic representation of the scent has been created in terms of an ordered set of elements, an emitter which contains chemical components corresponding to the ordered set of elements (e.g., receptor primary scent components; the elements of a Hedonic Vector Reference Set) can be used to re-create the scent, by activating chemical components within the emitter. If the ordered set of elements are linear in relationship, then re-creation of the scent is performed by linearly mixing the chemical components corresponding to the ordered set of elements. This permits substantially predictable reproduction of any scent by selecting the elements in proportional quantities. The ordered set of elements can be a set of scent components relative to a mammalian olfactory system, based on selective binding of chemical components making up the scent to olfactory receptors.

Preferably, the elements of the ordered set form an olfactory space or an olfactory subspace, for representation and re-creation of all scents or a specific set or subset of scents.

The electronic representation of the scent can include information about the relative intensity of the scent components required to reproduce the scent.

The electronic representation for re-creation of the scents can be generated as described herein. Alternatively, other users can generate scent representations, which can then be shared or transmitted via an information exchange network, an archive in a storage medium, such as a disk, or via the Internet.

Selection of Receptor Primary Scents by *in silico* Methods

An alternative method utilizes *in silico* screening techniques--that is, computer simulation methods--for either description of a sample scent, or for selection of candidate components to use as receptor primary, quasi-primary, or complex scent components, or for both description and selection. Protein-ligand screening can be used to select compounds which bind to particular receptors in order to identify receptor primary scent components. Examples of such programs are DOCK, AutoDock, GOLD, FlexX, LUDI, GROWMOL, and HOOK. (See Wang, J., Kollman, P.A., Kuntz I.D., "Flexible ligand docking: a multistep strategy approach," *Proteins* 36(1):1-19 (1999) and references therein.) These programs function by taking a protein structure and either matching compounds of known structure to the protein structure to determine the protein-ligand interaction, or by "growing" a molecule in the active site or binding site of a protein to determine what molecule will best interact with the protein. Interaction of a particular compound with an olfactory receptor can also be modeled in order to generate the information contained in the Receptor Docking Vector and associated fields.

Olfactory receptor proteins are membrane proteins, and experimental determination of the three-dimensional structures of membrane proteins has lagged the corresponding structural determination of water-soluble proteins for various reasons. However, alternative methods for constructing the three-dimensional structures of proteins are available. The primary (amino acid) sequences of many olfactory receptors are known. This information can be used to model a three-dimensional structure of a receptor protein using various algorithms and computer programs known in the art. The resulting model structure can then be used as the basis for evaluating interaction of candidate components with the receptor.

Alternatively, given known chemical structures which give rise to a particular odor, analysis of the structures can indicate the particular portion of the chemical structure which is responsible for the odor. This is analogous to "pharmacore analysis" used in medicinal chemistry to determine the important portion of drugs. Information generated by programs of

this type can be stored as the Structural Vector (and associated fields) in the Scent Object representation of a sample scent.

Methods for developing compounds which bind to receptors and other proteins of known structure, and determining interactions between ligands and receptors, are described in various references The DOCK program evaluates the fit of a ligand into a protein molecule of known structure (see Gschwend, D.A., Good, A.C. and Kuntz, I.D., "Molecular Docking Towards Drug Discovery", *J. Mol. Recognition* 9, 175-86 (1996); Kuntz, I.D., Meng, E.C., and B.K. Shoichet, "Structure-Based Strategies For Drug Design and Discovery", *Acc. Chem. Res.* 27, 117-123 (1994); and Kuntz, I.D., "Structure-based strategies for drug design and discovery", *Science* 257, 1078-1082 (1992); see also http://www.cmpharm.ucsf.edu/kuntz/dock.html). Using a known (or modeled) structure of an olfactory receptor, DOCK can be used to screen for compounds which bind to the receptor. The program AMBER (see Cornell, WD, Cieplak P, Bayly CI, Gould IR, Merz KM Jr, Ferguson DM, Spellmeyer DC, Fox T, Caldwell JW and Kollman PA. "A second generation force field for the simulation of proteins and nucleic acids," *Journal of the American Chemical Society* 117, 5179-5197 (1995); Computer Simulation of Biomolecular Systems, A. Wilkinson, P. Weiner, W. Van Gunsteren, eds. Volume 3, p. 83-96, P. Kollman, R. Dixon, W. Cornell, T. Fox, C. Chipot and A. Pohorille; Bayly CI, Cieplak P, Cornell WD and Kollman PA. "A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges - the RESP model," *Journal of Physical Chemistry* 97(40), 10269-10280 (1993); Cornell WD, Cieplak P, Bayly CI and Kollman PA. "Application of RESP charges to calculate conformational energies, hydrogen bond energies, and free energies of solvation," *Journal of the American Chemical Society* 115(21), 9620-9631 (1993); see also http://www.amber.ucsf.edu/amber/amber.html) can be used to calculate more precise interaction energies between candidate ligands. Other examples of such methods are described in, for example, U.S. Patent No. 5,866,343, directed to determining the energetically favorable binding site between two molecules; U.S. Patent No. 5,854,992, a system and method for structure-based drug design which takes into account binding free energy as it "grows" candidate molecules into a receptor binding site; and U.S. Patent No. 5,495,423, which describes a method for ligand design (principally applicable to peptidic ligands).

The foregoing methods typically depend on a known three-dimensional structure for the receptor. When such a structure cannot or has not been determined experimentally, a structure can be modeled using computer algorithms. Blundell TL, Sibanda BL, Sternberg MJ, Thornton JM, "Knowledge-based prediction of protein structures and the design of novel

molecules," *Nature* 326(6111):347-52 (1987); Shortle D, "Structure prediction: The state of the art," *Curr Biol* 9(6):R205-9 (1999), Morea V, Leplae R, Tramontano A, "Protein structure prediction and design," .*Biotechnol Annu Rev* 4:177-214 (1998) and Onuchic JN, Luthey-Schulten Z, Wolynes PG, "Theory of protein folding: the energy landscape perspective," *Annu Rev Phys Chem* 48:545-600 (1997) address various methods of predicting protein structure from sequence data. Various implementations for predicting protein structure from amino acid sequences are discussed in U.S. Patent Nos. 5,878,373 and 5,884,230.

If the structure, or even the identity, of the targeted receptor cannot be determined, alternative computational techniques can be used to generate information regarding possible ligands which will interact with the receptor. Quantitative structure-activity relationships (QSAR; see Green, S.M. and Marshall, G.R., "3-D QSAR: A current perspective," Trends Pharmacol Sci 16:285 (1995); and 3D QSAR in Drug Design: Theory, Methods and Applications, Kubinyi, H. Ed.; Escom, Leiden.), including QSAR refinements such as comparative molecular field analysis (ComFA) (Cramer, R. D. et al. "Comparative Molecular Field Analysis ComFA 1. Effect Of Shape On Binding Of Steroids To Carrier Proteins," *J. Am. Chem. Soc.* 110: 5959 (1988)); and pharmacophore mapping (Martin YC, Bures MG, Danaher EA, DeLazzer J, Lico I, Pavlik PA, "A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists," *J Comput Aided Mol Des* 7(1):83-102 (1993)) have been used to design pharmacophores that can interact with the receptor. U.S. Patent No. 5,699,268 provides a method for producing computer-simulated receptors which functionally mimic biological receptors; the simulated receptors are essentially abstractions of structurally useful information from compounds which are known to interact with a receptor. U.S. Patent No. 5,901,069 describes a method of automatically refining a set of chemicals using structure/activity data. U.S. Patent No. 5,862,514 describes a method of simulating synthesis of compounds of desired biological activity and evaluating their activity via further simulations.

Application of structure-function relationships to classification of odors has been described by Chastrette M., Rallet E. "Structure-minty odour relationships: Suggestion of an interaction pattern," *Flavour and Fragrance Journal*, 13(1):5-18 (1998); Chastrette M., De Saint Laumer J.Y.,; Peyraud J.F., "Adapting the structure of a neural network to extract chemical information. Application to structure-odour relationships," *SAR QSAR Environ Res* 1 (2-3):221-231 (1993), Chastrette M., "Trends in structure-odor relationships," *SAR QSAR Environ Res* 6(3-4):215-254 (1997) and Jain et al., "A shape-based machine learning tool for drug design," *J Comput Aided Mol Des* 8(6):635-652 (1994). These methods can be useful in

determining the "chemical distance" between odors. For example, isoamyl acetate is typically experienced as a banana-like odor, while octyl acetate is typically experienced as an orange-like odor, which gives a measure of how the chain length of the alkoxy portion of the ester influences perception. Example 3 describes a structural comparison of compounds within and across subjectively-defined scent categories.

Scent Reproduction/Approximation by Automated Iteration

Scents can also be reproduced or approximated by automated iteration and emission of candidate scents, which are then evaluated for their similarity to the sample scent. A scent can be sampled by a human operator, who can input certain characteristics of the scent (e.g., "major characteristic: pungent; minor characteristic: citrus"). The system can then select various odorants and emit appropriate combinations of them as candidate scents for comparison against the reference scent. While the human operator can rate the candidate scents in terms of how similar or dissimilar they are to the sample scent, other methods to rate the candidate scents against the sample scent can also be used, such as the response of an "artificial nose" (an artificial scent-sensing device), or the output of a gas chromatograph. Alternatively, an automatic method, such as the response generated by an "artificial nose" (an artificial scent-sensing device) can be used in the initial sampling step as well. Various combinations of scents are used in order to minimize the difference in response (whether of a human operator or an automatic element) between the sample scent and the candidate scent. Algorithms for exploring such multivariable spaces are well-known, as the problem essentially reduces to minimizing the difference between the response function, F, to the sample scent and the candidate scent; that is, finding the closest solution to

$$F(\text{sample scent}) - F(\text{candidate scent}) = 0.$$

Multivariable minimization techniques can often become trapped in "relative minima;" methods for compensating for this problem are well-known in the art.

Scent Fingerprinting

It will be appreciated that in many instances, analysis of a scent (whether expressed in terms of receptor primary scent components, receptor quasi-primary scent components, receptor complex scent components, perceptive primary scent components, perceptive quasi-primary scent components, perceptive complex scent components,or other scent representations) is of great utility in and of itself, in addition to the utility of that analysis in scent re-creation. Thus, another embodiment of the invention encompasses "scent

fingerprinting," which comprises analysis of a scent profile when re-creation of that scent may not be necessary or desirable. The distinction between scent profiling, as defined above, and scent fingerprinting, as defined here, is that scent profiling is a representation of a scent relative to a mammalian olfactory system in such a manner as to provide useful information about the interaction of the scent with that olfactory system, such as sufficient information to enable re-creation of the scent from receptor primary scent components, or information about the response of one or more humans to the scent to enable re-creation of the scent from perceptive primary scent components, perceptive quasi-primary scent components, and/or perceptive complex scent components. In contrast, scent fingerprinting can, but does not necessarily, provide such information.

Various applications and examples of scent fingerprinting can include, but are not limited to, the following illustrative situations. Natural gas is widely used as a heating and fuel supply, but is in itself odorless. Utility companies routinely add small amounts of odorants such as mercaptans to allow detection of natural gas leaks in households. Should a leak occur at an unattended site, however, potentially dangerous quantities of natural gas can accumulate. In such areas, a device which can recognize odorants would be useful.

Another use of scent fingerprinting is quality control of a manufacturing process. Many food items, such as freshly-baked bread and pastries, sauces, and cheeses, have distinct odors. A manufacturer can record a scent fingerprint for a given food item, e.g. spaghetti sauce for packaging in jars. The quality of the product can then be monitored at various stages in manufacture and storage, and deviations from the established scent fingerprint can be used to alert the manufacturer to problems in manufacture or storage. Quality control scent fingerprints are not limited to food items, but can be used in any circumstance where a volatile component of an item of manufacture can be used as a quality control indicator, e.g., perfume, deodorants, gasoline and other fuels, solvent mixtures, etc.

While scent fingerprints need not be meaningful in terms of a mammalian olfactory system, it will be readily appreciated that a scent profile, which does represent a scent in a manner relevant to an olfactory system, is a special type of scent fingerprint. Additionally, the response of an artificial scent-sensing device which yields a scent fingerprint of an odor, such as the "artificial nose" described in U.S. Pat. Nos. 5,571,401, 5,698,089, 5,788,833, 5,891,398 and 5,911,872, can be measured and stored as the Polymeric Reactivity Vector and associated fields of a Scent Object. The response of the device can be calibrated against the response of a mammalian olfactory system. This known relationship can then be used to transform the scent fingerprint generated by the device into a true scent profile. In turn, the scent profile generated

can be utilized to re-create an odor using receptor primary scent components, receptor quasi-primary scent components, receptor complex scent components, perceptive primary scent components, perceptive quasi-primary scent components, and/or perceptive complex scent components. The invention encompasses such transformations.

Recording the Analyzed Scent in Electronic Form

The scent which is analyzed by the analysis module can be recorded in electronic form during or after the performance of the analysis. The exact nature of the electronic representation of the scent will initially be dependent on the manner in which it is analyzed. For example, a scent which is analyzed by exposure to a panel of olfactory receptors can be represented as a multicomponent Receptor Screening Vector as described above, where each component of the vector comprises a measure of interaction with the individual receptor, while a scent which is exposed to an artificial scent-sensing device ("artificial nose," e.g. of U.S. Patent No. 5,911,872) can be represented in terms of the interaction of the scent with each element of the device as a Polymeric Reactivity Vector. However, one embodiment of the invention embraces transformation of the representation of a scent from one form to another. Thus, a representation of a scent by a device which has no correspondence to natural olfactory receptors can be transformed into a representation that does correspond to natural olfactory receptors. If the scents are represented by vectors, this can be accomplished by constructing an $n \times m$ matrix (where the number of rows, n, corresponds to the number of elements of the vector being transformed, and the number of columns, m, corresponds to the number of elements of the vector resulting from the transformation). Vectors for representations of scents can be readily stored as array data structures on electronic computers; arrays of arrays can be used to represent more complicated scents.

One particularly useful representation of data is in terms of index values to be used for the emitter module. The index values consist of the data passed to the emitter which enable re-creation of the scent by the emitter. The various scent components (whether receptor primary, quasi-primary, complex, or perceptive primary or complex) available at the emitter are mixed according to the index values. For a scent which has been represented in terms of all receptor primary scent components, which is to be re-created by an emitter which can mix primary scents corresponding to all receptor primary scent components, the representation of the scent as the primary scent component vector is equivalent to representation as index values. For the example of the six olfactory receptor scent, (1, 1, 1, 1, 1, 1), where two receptor complex scent components (1, 0, 0, 0, 1, 0) and (0, 1, 1, 1, 0, 1) are available, the index values for the two

receptor complex scent components would be 1 and 1 (equal mixing of the two components to re-create the scent). The original scent profile of the scent, with respect to the olfactory receptors, describes the scent in terms of the analytical system used for profiling, while the index values for the scent describe the scent in terms of the chemicals available at the emitter module for scent re-creation. The two representations are equivalent and can be interconverted. Either representation is suitable for transmission over a data network, in digital, electronic, or other form, or for recording in storage media, including, but not limited to, computer disks, computer tapes, printed formats, or RAM, ROM, PROM, and EPROM computer memory devices. In the Scent Object described above, the index values form the Palette Vector, indicating which chemicals from the Scent Palette should be combined, and in what ratio, to re-create a scent.

Scent Editing

Representation of a scent as a scent profile provides the capability of editing the scent. A scent profile which represents a scent in terms of perceptive primary scent components is the most straightforward representation to edit. An example is the perceptive complex primary scent of "burned pizza" comprised of perceptive primary scent components of sausage, cheese, tomato sauce, and burned dough. In order to edit the scent to provide a more pleasant re-creation, the perceptive primary scent component of burned dough would simply be eliminated. The perceptive primary scent component representation corresponds most closely to the Hedonic Vector representation of a scent in the Scent Object. Manipulation of the Hedonic Values of a particular scent object provides a straightforward means of scent editing. Manipulation of the Palette Vector values of a particular scent object also provides a ready means of editing the scent. An example of scent editing is provided in Example 1.

Other scent profiles can be edited using a knowledge of the perception of a particular components. Using our six-receptor example, suppose that the (1, 0, 0, 0, 1, 0) receptor complex scent component is known to provide an unpleasant aspect of the scent, while the (0, 1, 1, 1, 0, 1) component is known to provide the pleasant aspect of the scent. The first complex scent component can be omitted from the edited scent profile, leaving (0, 1, 1, 1, 0, 1) as the edited scent profile. (This would also alter the index values for scent re-creation, from 1 and 1, to 0 and 1.) More complex editing situations can be manipulated using computer algorithms as discussed above.

Individual scent components can be omitted, added, weakened, or intensified, and different scent components can be adjusted in different manners or degrees, depending on the

desired result. The editing can be done interactively, with each edited scent emitted by the emitter module for evaluation by the user, or can be done automatically, with removal/weakening or addition/intensifying of particular components specified in advance, on either an absolute scale or relative to other components.

Scents can also be created, by selecting two or more elements from an ordered set of elements that represents or corresponds to an olfactory space and creating a scent profile from the elements. The scent can then be emitted by an emitting device in the same manner as any other scent expressed a electronic representation. The relative intensity of the elements can be varied, in order to adjust the resulting scent; preferably, the elements are related linearly, so that changes in the electronic profile cause linear changes in the resulting scent. For example, if it is desired to activate two scent receptors in a mammalian olfactory receptor system, multiplying the value for one element by a constant should increase the intensity of stimulation of the olfactory receptor by the same constant. A created scent can also be edited by any other method described herein.

The scents may also be created in terms of the index values of the scents available at the emitter device.

Emission Subsystem

The emission subsystem is essentially a "playback" unit for the scent as represented by the object file. Data from the object file can be compressed, uncompressed, or streamed over the Internet and is used to activate the appropriate chemicals which correspond to the index values in appropriate concentrations to reconstruct the scent. For example, the emission subsystem may comprise cartridges of proprietary chemical reagents in an inkjet printer like device. The corresponding reagents in the appropriate amount are sprayed onto an emission medium, where they are blended and vaporized to re-create the scent. The scent chemical cartridge can be a general cartridge, containing a number of reagents which may be dispensed to generate all scents, or as many scents as practicable. The scent chemical cartridge can also be a specialized cartridge, intended to hold a limited subset of chemicals which are well-suited for generating limited scents appropriate and relevant to a specific application (e.g., a perfume cartridge may have chemicals useful in generating a wide type of perfume scents; a game cartridge may have a limited set of chemicals useful in generating scents appropriate to an associated game such as vehicular scents for a racing game, munitions scents for a war game, etc.). The emission system can be capable of using one cartridge, two cartridges, or more than two cartridges; for example, a dual-cartridge emission system can contain a "general" scent

cartridge, which contains odorant substances suitable for reproducing common scents, and a "specialty" cartridge, which contains odorant substances suitable for reproducing scents limited to a specific context or application. The operating system for the scent playback unit should be capable of distinguishing between cartridges and adjusting its output accordingly, that is, the system can determine what odorant substances are available, how much of them are to be dispensed, and how to mix them accordingly.

Schematically, FIG. 2 illustrates an alternate embodiment of an emitter subsystem in which a deep well plate emission system 60 is used to store the scent reagents. A deep well plate 62 contains the reagents needed to provide a particular scent. The capillaries 64 carry the reagents through a computer controllable well-specific microvalves 66. The microvalves 66 are linked to a controller 68 which determines the flow of the reagents to the corridor 70 where heating and mixing of component odorants occur. The corridor 70 is either a liquid or air flow chamber from which the vapors 72 give off the desired scent.

Other emitters include a resistor powered heating plate for boiling off the scent chemicals. In a system 80 depicted in FIG. 3A, a small portion of the reservoir 82 of the liquid is heated via a resistive element 84, causing an amount of the liquid to vaporize and a droplet is formed in a syringe or capillary 86 or other depositing tube by condensation, and the expanding vapor in the reservoir forces the droplet out of the tube to drop onto a heated evaporation site 88. The heat from this evaporation site 88 accelerates evaporation of the liquid into vapor to be transported by a stream of moving air 90, which flows out through the front of the device to the user. Residual deposits from incomplete evaporation may contaminate the evaporation site 88 and consequently may also contaminate subsequent intended scents. In a more elaborate scheme, it may be desirable to heat the evaporation site with different temperatures at different regions to provide different rates of evaporation. Alternatively, instead of heating the liquid, the air space in contact with the liquid can be heated; the resulting vapor pressure of the heated air can be used to push the liquid through the syringe, capillary, or other depositing tube. Functions and operations of the various components are controlled and synchronized by the controller 92 in accordance with the input signal representing the scents to be played back.

In a system 100 depicted in FIG. 3B, much like in FIG. 3A, a small portion of the reservoir 102 of the liquid is heated via a resistive element 104, causing an amount of the liquid to evaporate and a droplet is formed in a syringe or capillary 106 or other depositing tube by condensation, and the expanding vapor in the reservoir forces the droplet out of the tube to drop onto a heated evaporation site 108. The unique aspect of this evaporation site 108

is the fact that each pre-defined region has its own resistive heating element 112. Thus, each region has its own rate of evaporation of the liquid into vapor to be transported by a stream of moving air 110, which flows out through the front of the device to the user. Functions and operations of the various components are controlled and synchronized by the controller 114 in accordance with the input signal representing the scents to be played back.

In the preferred embodiment of FIG. 4, a rolling blotter paper 122 over a stationary heating element 124 combines a simple and reliable solution to the serious problem of residual contamination of the air stream. A liquid droplet 126 drops onto a sheet of blotter paper 122, which then rolls over a stationary curved heating surface 124 for evaporation. After the appropriate time is allotted for evaporation, the sheet continues to roll into a take-up wheel 128, where there is no longer any heat applied, and consequently the scent of the residue on this section of sheet will be stifled. The heated surface 124 will be constantly cleaned by the natural motion of the paper over it, as more and more scents are played by the user. The flow of air 130 over the heated element 124 will carry the scent to the user. Functions and operations of the various components are controlled and synchronized by the controller 132 in accordance with the input signal representing the scents to be played back.

The replacement of the blotter paper will be designed to reflect usability on the same level as replacing printer paper in an electronic adding machine, cash register, or other similar device which uses a rolling paper mechanism. Because this design employs continuous movement of the paper during the scent-playback mode, the life of the roll of blotter paper will depend on the number of wells of liquid as well as the speed with which the paper moves through the emitter system. The purely mechanical approach to removing scent residue (as opposed to relying on the fluid mechanics and timing of evaporation) assures a reliable ceasing of residual odor and predictable functionality. This reliable nature addresses the sharp "on" and "off" which may be desired by the user of the scents. In an alternative embodiment, the rolling blotter paper design may employ a mechanism in which the paper stops moving while the liquid droplets are being deposited. In this manner, the blotter paper will once again resume rolling until a need for another scent is requested. (The scents "played" by this and other emitters are sometimes referred to as "scent notes.") This non-continuous rolling of blotter paper will repeat until all scent notes have been played back.

Another feature of the blotter paper method is the ability to fade smells in and out and perhaps also alter blends of smells. Much like a player piano, the blotter paper will roll along, accepting liquids as it goes. The device can begin by squirting one liquid out onto the paper,

while slowly fading in subsequent smells in order to create blends. Also, pre-existing blends may also be altered, depending on the proximity of the release of one liquid after the other.

In another embodiment depicted in FIG. 5, the system 140 has individual heated pads 144 mounted directly below the orifice 150 through which the liquid 142 emerges after boiling off by the resistor 148 in the reservoir. The liquid wicks out onto the surface of the heated pad 144 outside the reservoir and then evaporates into the air stream 152 out to the user. Since these heated pads 144 are individually controlled, the heat to a previous scent note may be stopped before heat to a subsequent note is applied. A small amount of reagent residue from a previous scent note played may remain on the heated pad for that note. The amount of contamination by the residual of an earlier scent note is dependent in part on the control of the cooling of the pads. The amount of liquid remaining on the heated pad after it is shut off will depend in part on how well such liquid wick. Uniform wicking is desired as the control of the evaporation would be more predictable. The tracking accuracy of the scent emission from the emitter in synchronization with media presentation depends in part on the length of time it takes for an individual pad to heat up to sufficient temperatures, as well as to cool down sufficiently so that it no longer promotes evaporation. The functions and operations of the various components are controlled and synchronized by the controller 154 in accordance with the input signal representing the scents to be played back. The heated pads may be an integral part of a disposable/replaceable cartridge unit that contains the scent reagent reservoirs. The heated pads would be disposed of with the cartridge. The cost of the heated pad may be kept to a minimum by adopting conventional printed circuit or deposition techniques to form the pads. It is important to note that in any of the aforementioned systems 80, 100, 120, and 140, a reversible fan can be incorporated relatively easily into airflow channels of the embodiments and may help solve the problem of sharp "on" and "off" of desired scents.

Other emitter designs involve the fluid mechanical properties of the moving air used in the device. In FIG. 6, the system 160 takes advantage of the low pressure created by a jet of high speed air 162 through a Venturi tube 164. This design would require a separate Venturi tube 164 to be set up for each individual well of liquid 166, which is regulated by a valve 168. When the desired scent note is requested, the corresponding valve 168 opens and allows some of the vapor from the reservoir 170 to be drawn into the fast moving air stream 162 outside via the low pressure induced by the Venturi tube. This vapor would then flow with the air out through the front of the device to the user. Functions and operations of the various components are controlled and synchronized by the controller 172 in accordance with the input signal representing the scents to be played back.

As depicted in FIG. 7, yet another embodiment of the emitter 180 involves a high-surface area medium, such as a gauze 182 (which can be cotton gauze, silk gauze, or made from any other suitable material), to be doused in the liquid 184. This medium would be contained in its own reservoir or compartment 186, sealed off with a valve 188. When the desired scent note is requested, the valve 188 opens, and a jet of air 190 blows through the gauzy surface, blowing evaporated gas and/or reagents sitting in the reservoir, along with perhaps some fine particles of liquid into the transverse air stream above. This combination of gas, liquid and air would then travel through the front of the device to the user. Functions and operations of the various components are controlled and synchronized by the controller 192 in accordance with the input signal representing the scents to be played back. A benefit of this approach is the relative simplicity in the method of expulsion. As the gauze only needs to be exposed to a pressurized air stream, no additional heating elements or circuitry are needed in order for the scent note to be expelled into the air stream

Another emitter design involves the squirting of jets of chemical reagents representing primary scents out toward the user, a space (e.g., a vaporization chamber or corridor as in Fig. 2) or a medium (e.g., a piece of blotting paper) for vaporization. The control of the liquid jet is analogous and may be referenced to a Hewlett Packard Ink Jet printer (e.g., U.S. Patent No. 4,532,530). The reagents on the blotting paper may be vaporized in similar manners described in the above embodiments (e.g., Fig. 2 and Fig. 4. Usage of the above-described embodiments would depend in part on the balance between the value of the specific applications versus the cost of replacement of the disposable components.

Instead of using liquid reagents, reagents in solid phase may be used as the starting material for generating scents. The solid reagents may be in the form of palettes or paste stored in receptacles such as containers or wells. The solid reagents are melted and vaporized selectively, such as using the afore-described methods to obtain the desired combination of primary scents. Reagents may be developed such that they do not emit odor when they are in the solid phase. Further, solid reagents may be developed that sublime from solid phase to gaseous phase. Such solid reagents are easier to handle and process, and the emitter design may be simplified.

Another alternative is the use of tanks or other containers of gaseous substances. Regulators can be controlled via computers or other electronic or electromechanical devices to emit appropriate volumes of gas.

Alternatively, the reagents may be impregnated into substrate based carriers, such as beads, gauze or other substrates that offer exposure of the reagent to a large surface area for vaporization to generate the scents.

Other scent-producing members may be adapted to emit scent in the emitter subsystem as well. U.S. Patent No. 5,887,118, which is fully incorporated by reference herein, discloses a scent-producing member with an associated resistive element to act as triggers for scent production. Also, mechanisms such as scent rolls as disclosed in U.S. Patent No. 5,832,320, which is fully incorporated by reference herein, may be employed to provide the odorants by rolling the desired scents onto a scent carrier. Other emitting devices such as a heating plate and a fan which upon actuation causes perfume in the emitting device to be release can be used as disclosed in U.S. Patent No. 5,398,070, which is fully incorporated by reference herein. Further examples of existing emitters are described in U.S. Patent No. 5,591,409, which is fully incorporated by reference herein, where metered spray technology introduces controlled amounts of aromatic chemicals in the air from spray containers. Similarly, U.S. Patent No. 5,724,256, which is fully incorporated by reference herein, discloses a odor dispensing system in which a linear array of odorant containers dispense odorants onto a rotating absorptive porous member connected to an air channeling device.

In all of the above embodiments, a carrier can be mixed or otherwise combined with the odorant substances for emission. The carrier is preferably inert (non-reactive with the odorant substances) and odorless. The carrier is selected for compatibility with the form of the substances used; for example, liquid reagents can be mixed with another liquid, such as water, and gaseous reagents can be mixed with another gas, such as ambient room air, a standard room air mixture contained in a tank, helium, or other gases. The carrier can be combined with the odorants at any stage of the emission process, e.g., the carrier can be combined with odorant substances immediately before emission; or it can be combined with each odorant substance individually followed by combination of the odorant-carrier mixtures and then emission of the combined mixtures; or it can be combined with one odorant, followed by addition of more odorants to the initial odorant-carrier mixture; or it can be combined with mixtures of odorants at any stage prior to emission.

The subsystems may find utility and embodied in various implementations without departing from the scope and spirit of the invention, as will be apparent from an understanding of the principles that underlie the invention. It is understood that the emitter concept of the present invention may be embodied in different types of hardware and/or software platforms, whether in an information exchange network environment or otherwise. For example, the

present invention may be applied to devices for game, business, productivity and other types of applications.

It will be appreciated that the line between hardware and software is not always sharp, and those skilled in the art will understand that such networks and communications facility involve both software and hardware aspects. A method or process is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. These steps require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

Useful devices for performing the operations of the present invention include, but is not limited to, general or specific purpose electronic processing and/or information processing devices, which devices may be standalone devices or part of a larger system. As used in the context of the present invention, and generally, electronic processing and information processing systems may include computers (such as personal computers), personal digital assistant, video game consoles, television consoles, set-top boxes, broadcast radio consoles, application specific systems, and other systems and devices that incorporates a processing unit. The devices may be selectively activated or reconfigured by a program, routine and/or a sequence of instructions and/or logic stored in the devices. Memory devices, which can be used for memory modules of the system, include, but are not limited to, videotapes, audiotapes, cassette tapes, compact disks (CD's), digital video disks or digital versatile disks (DVD's), computer memory devices such as ROM, RAM, PROM, EPROM, hard disks, floppy disks, electronic files, software cartridges, Web pages, file servers, and the like. Control devices which can be used in the system include, but are not limited to, microprocessors, integrated circuits, general purpose computers, and special purpose computers. In short, use of the concepts described and suggested herein is not limited to a particular processing configuration.

The subsystems mentioned above are preferably software driven. The operating system and programming language, protocols, and drivers are developed for the various system components. Preferably, the operating system, protocols and drivers should be hardware and/or software platform independent.

In a presently preferred embodiment of the invention, scent objects may be incorporated into Internet or computer software applications, such as multimedia applications or content deployed over the World Wide Web, through the use of functions supplied from a software development kit (SDK) and executable on a runtime machine, such as a personal computer. In one embodiment of the present invention, the SDK is a COM model object packaged as a dynamically linked library (DLL) for use as an in process server. Preferably, the SDK may be handled as a standard COM object running under Microsoft Windows, exporting the requisite functions to the application programs, although the functions of the SDK may be written in C++ source code or other programming language known in the art, and may be platform and operating system independent. In addition to the SDK, a low level device driver is also installed on the runtime machine to interface between the SDK functions used in the applications programs and the scent emitting device in a manner well known in the art Preferably, the SDK DLL should be included with the software applications that incorporate SDK functions, and should make use of the scent emitting device device driver present on the runtime machine. If there is no scent emitting device and/or corresponding device driver installed on the runtime machine, the SDK functions should fail gracefully, without affecting the exection of other portions of the application program.

Preferably, the scent objects may be proliferated via the Internet and may include a built-in security key. The SDK will not emit a scent unless the corresponding scent object contains a valid key. If the key is invalid, the function emitScent() will fail and issue and ssInvalidKey error message. Scents that do not contain keys are called unauthorized scents. Scents with embedded keys are authorized scents. Unauthorized scents may be emitted using a scent browser as described above. Preferably, the SDK may be distributed with a set of fully authorized dummy scent objects containing valid keys. These dummy scent objects may be used for testing with an emulator program, discussed below, and should be replaced with actual scent objects defined in the manner described above for use with a scent emitter device.

The SDK has a simple flow of execution. At start run the object is created using a helper function, such as IScentWareSDK *CreateScentWare(void). At the end of program execution the SDK object is destroyed once again with a boilerplate helper function, such as

ssBool DestroyScentWareSDK(IScentWareSDK *theScentWare SDK). This function simply calls the SDK's Release() method.

Once the object has been created the environment may be opened and closed using the open() and close() member functions. After a successful call to open() has been made, it is possible to call one of the emitScent(...) function signatures or the accessor and mutator functions of the ScentWare object. When the application is finished using the SDK, a call is made to close(). In addition, the SDK supplies a function to allow the software environment to set the communications port of an scent emitter device prior to the call to open(). Version accessor functions will allow the software application to obtain version information on both the SDK as well as the scent emitter device driver on the runtime machine. If no driver is found then the getDriverVersion() function will return NULL. A function isDevicePresent() may also be defined to determine if a scent emitter device is present on the runtime machine, thus permitting the application program incorporating the function to enable or disable any preference controls related to scent.

The SDK runs on top of a thread that does the work of emitting a scent and timing its persistence envelope. This thread has a lower than normal priority. The rendering thread also prevents the emission of a given scent if the same scent is still persistent. In addition, if an emit call is made for a different scent while the scent emitter device while the device is actually emitting scent essence, the scent will be held pending until the device is not busy and the scent will be emitted. It is only possible to hold a single scent pending at any given time.

In one embodiment of the present invention, a scent emitter device may operate by emitting small portions of scent essences. A vaporizer may vaporize the essence mixture, which is then diffused into the surrounding air by the diffuser. Vaporization refers to the amount of heat applied to the scent essence in a vaporization chamber in the scent emitting device. Preferably, vaporization is represented in the SDK as a number from 0 to a maximum vaporizer setting, where 0 is set to off. Higher vaporization values will create scents with a sharp onset and shorter persistence, as scents will be vaporized more quickly. Diffusion refers to the amount of air passed through a scent vaporization chamber. In the software functions of

the SDK, diffusion is also represented as a number from 0 to some maximum diffusion value, such as 255. Lower diffusion rates will soften the strength of the scent.

Preferably, each scent has vaporizer and diffuser settings in the object, so that it is not necessary to specifically call setVaporizer() or setDiffuser(). An overloaded signature of emitScent() may also be available to allow the calling application full control of these settings.

As discussed above, the intensity of a scent is based on several factors, the foremost of which is the amount of essence that is emitted. Vaporizer and diffuser settings can also affect the intensity but to a lesser degree than the amount of essence emitted. The SDK settings for intensity deal with the amount of essence emitted by the scent emitter device. In addition, as discussed above, a scent can be either simple or complex; in one embodiment of the present invention, a simple scent uses only a single scent essence while a complex scent uses up to eight different essences. Complex scents may be interpreted in the SDK as a product of several different essences emitted with varying intensities. Preferably, both types of scents preferably have singular Vaporization and Diffuser settings for their intensity level, ssMild, ssNormal, and ssStrong.

As discussed above, a scent has two phases of existence: emission and persistence. While the scent is being emitted by the scent emitter device, the device is busy and cannot accept further commands. If a command for a dissimilar scent to the current scent is received during emission time, it should be held pending. While the scent is persistent in the air, the thread mechanism disallows the smell emitter device from producing further emissions of the same scent. Functions may be written to access scent objects of recently emitted scents in order to determine whether a given scent, or even any scent is currently persistent. The thread mechanism prevents a succession of game space object triggers from emitting an overwhelming scent when only a single emission is needed. Iteratively calling emitScent() without the protection of the threads persistence aging process can overload the vaporizer and produce a strong scent with an overly long persistence envelope.

In one embodiment of the present invention, the command slot for the thread is a single command deep. If a command is already pending, successive calls will fail. When the thread accepts the command it clears the command slot and makes it ready for the next command.

In a scent enabling an interactive application, generally, "less is more". A world scent that is triggered every 5-10 minutes coupled with prize, event, or entity based scents is one potential design pattern. Because of the nature of scent persistence, the calling application should be discreet about the number of calls made. The thread based protection from emitting multiple iterations of the same scent, as discussed above, will offer further protection from excessive triggering of scents.

In a preferred embodiment of the present invention, a software emulator executable in a conventional computer operating system, such as Microsoft Windows, may be supplied. The software emulator, when connected to a host station via a null modem cable, will log and display all transactions with a virtual scent emitting device located at the host station for the purposes of development testing during the process of scent enabling software applications. The emulator requires a second computer (the target) to be connected to the host (development) station via the null modem. Preferably, this cable should be connected from communications port COM1 on the host to COM1 on the target. Once the connection is made, the emulator program may be run on the target machine, with a software application that is scent enabled using the SDK function executing on the host station. The emulator outputs the commands sent to it as text in a text window. The emulator program may use a list of scent names supplied in a text file that may be edited in order to customize the list of scent names as needed. Preferably, the file format is a simple text file containing the names of each scent on a separate line with the word End followed by a return at the end of the listing.

The Appendix contains source code, including function prototypes, written in the C++ programming language, for a computer implementation of the SDK.

Applications of Technology

It is evident that the ability to record, store, transmit, and playback electronic scents would find many applications. For example, scents may be stored for forensic purposes. This storage system would allow for quick dissemination of information between various investigative agencies and would eliminate cumbersome storage facilities. In other applications, scents for food or cosmetic products may be stored as standards for quality

control of production. The olfactory characteristics of the production would be measured against that of the stored standards.

In other commercial embodiments, a electronic "scent signature" can be created for advertising or other purposes. A particular stored scent profile can be associated with a given location, event, person, object, concept, or product. Retail stores such as restaurants, coffee shops, and other variety shops may wish to broadcast a particular scent profile along with visual and audio advertising. The present invention will be especially useful with the expanded use of the internet and related computer applications. Online sampling of scents will make available for consumers to evaluate new items before deciding on a particular purchase. Online shoppers (e.g., for perfume) could now sample the scent of the product when making a purchase decision. Such applications would add a new dimension of virtual reality to advertisements. Association of a scent representation, which can be conveniently stored in a file on a computer, with daily events, advertising, promotion, products, services, content delivery, hardware operating status, thought, location, physical objects, tangible objects, and intangible objects are additional applications. For example, content delivery consisting of movies can contain scents appropriate to scenes in the movie, such as floral scents during a depiction of a garden scene. A Web page displaying a location such as a beach could generate scents associated with that location. Electronic greeting cards can be customized with scent, such as the addition of the scent of flowers to an electronic Valentine's Day card.

In the context of multimedia presentations, the electronic scent data may be embedded in a multimedia file to synchronize the production of scents along with visual and audio presentation of the multimedia content. For example, virtual reality games could now include scent for more realistic game play. Adult entertainment can be enhanced by specific scents. Scents can be embedded into CD's, DVD's, movies, or broadcast with television and radio programs. It is important to note that with the present invention, it will be possible to also "dub" the scents onto existing CD's and DVD's as an aftermarket feature. For cable or satellite TV broadcasts, existing tracks for closed captioning may be used to "piggyback" the transmission of electronic scents in the programs. Thus, the proper ambience for each movie, song, or game will be provided with the desirable scents. A system for streaming scent playback analogous to the RealAudio/RealNetworks "streaming" method for audio may be desirable. For broadcast radio, electronic scent data may be transmitted in a complementary frequency or multiplexed fashion in synchronization with the radio programming to enable the listener to experience scents that correspond to the particular radio programming. The scents

may be playback with an emitter that is integrated with the radio or coupled to the radio frequency.

Scent emission could also be synchronized to the operating activities of a particular electronic system. Personal computers, set-top boxes, audio systems, on-board computers on automobiles, central monitoring unit in a home, as well as other systems could be programmed to emit certain scents based on the then current function of the system. For example, a certain scent could be matched up with a computer's initial boot-up sequence, while another scent could go with an error message, and another for shut-down mode. Furthermore, scents of these systems could be synchronized with the times of the day. For example, a personal computer, a set-top box connected to a TV, or a central monitoring unit of a home could emit the smell of coffee and fresh-cut flowers during the morning hours and send off the smell of a fireplace or a barbecue in the evening.

Other forms of advertising via the World Wide Web on the internet are possible. For example, banner advertising which are now prevalent throughout various web sites can utilize the scent technology. In an effort to grasp people's attention, banner advertisements for flower shops, for example, could emit floral scents either automatically as soon as the banner appears on the screen, or upon the user's clicking on the banner itself. For banners which emit scents automatically, the banner ad itself would have the scent object files incorporated in the banner along with other advertising information. For click-and-smell banners, the banners could simply employ a hyperlink which would reference the scent object files. A particular user's clicking of the hyperlink would initiate the scent transmission and the server web site would "serve" the scent object file. In effect, the "click-and-smell" function would simulate a "scratch-and-sniff" motif found in various print advertisements of popular magazines. With this new scent feature for banners, the advertisements can "come alive" and add a more forceful promotional effect. User interactive games in which people mail scent objects to each other will also be possible. Scent enabled greeting cards and electronic mail will also provide further marketing tools. In such as case, the greeting cards as well as the electronic mail will be sent from one user to another and will contain the scent chosen by the sender.

Fig. 13 is a pictorial representation of the sequence of scent digitization into a electronic file, broadcasting of the electronic file , and synthesis of the primary scent based on the electronic file.

Electronic encoding of scents also gives flexibility to create new scents. In particular, by providing a unique programming language for the creation of scents, one can design new scents electronically, analogous to computer-aided design of physical or musical objects.

Thus, a fine tuning of the desired scents can be attained for maximum satisfaction of the users. Existing scents can also be edited to add or improve on desirable characteristics, or to remove unwanted characteristics. Consumer products (e.g., shampoos) can be reformulated for more pleasant odor characteristics. One particularly important application of scent editing is in flavoring. Taste perception is strongly odor-dependent, and the flavor of foods can be significantly improved by editing a scent profile of a given food, followed by the addition of an odor to enhance a particular desired flavor. By the same method, the use of a scent-blocking additive can block a particular undesired flavor.

Commercial Applications of Scent Technologies

Scent Databases

A web site would provide end users such as advertisers a depository/database of scents to be tested to create various new scents. Table 1 of Example 1 provides a description of one such initial database; as new scents and new combinations are created, the scent objects corresponding with the newly-created scent is registered in the scent database. Users can sign up with the web site and each receive a unique user identification and a password to access the system. In the system, each scent object would have a textual description as well as a hyperlink for end users to read and explore, respectively. One model which could be used to classify the various scents for end users is a wine aroma wheel as depicted in A.C. Noble, R.A. Arnold, J. Buechsenstein, E. J. Leach, J.O. Schmidy, and P.M. Stern, "Modification of a standardized system of wine aroma terminology", *American Journal of Enology and Viticulture*, 38/2 (1987). The wine aroma wheel would provide an existing "platform" of scents based on pre-classified and well-known aromas. Another way of classifying and presenting scents would be to employ a list of available scents. In either scenario scents would be sub-grouped into categories and the end users could browse the web site to sample the scents.

Advertisers or content developers will be able to purchase electronic objects, typically scent objects (e.g., online over an information exchange network such as the Internet) so that their respective applications may take advantage of this electronic scent technology. As mentioned above, these electronic objects available for purchase will be categorized based on scent classes and form a registry of electronic scent objects. For example, a beach hot dog vendor may have a web site promoting his hot dog stand on the internet. As such, the vendor may wish to incorporate the smell of hot dogs with his advertisement on the World Wide Web. The vendor could license a electronic description with the generic hot dog smell. A database

of scent objects--in effect, a "scent registry"--can be created, with various scents of interest to vendors, advertisers, and individuals or groups which want to incorporate scent into their Web page or other presentation (such as email). The database can be accessed, sampled, and a scent object can be purchased or licensed for placement on the Web page or other presentation. In one embodiment, the database is placed on a Web site, with appropriate text and other descriptors of scents. For example, a scent object entry for decanoic acid can state: "Decanoic Acid. Scent characteristics include fatty, rancid, citrus. Food/beverage compatibility: appears on United States Food and Drug Administration list of 'generally recognized as safe' food additives." In practice, the scent objects listed will be for complex scents, with many elements. The scent objects can be listed, or sorted, by the perception of the scent (e.g., "citrus"), by the recommended application of the scent object or recommended applications to avoid (e.g., "best used in perfumes; suitable for lotions; cannot be used in detergents"), by its toxicological data (e.g., "safe for skin contact; not safe for ingestion"), by its cost, by its principal or secondary chemical components, or by any other characteristic that would affect the use of the scent corresponding to that scent object.

The registry can incorporate such features as: checking newly-created scent objects against a database existing scent objects, to determine whether the newly-created scent object has already been created by another user (or whether an existing scent object is similar to the newly-created scent object, and the degree of similarity, which can be based on the perceptive classification of the scent, the chemical recipe for the scent, or other indices); checking the newly-created scent object against a database to screen for possible infringement of patent or other intellectual property rights; or checking the newly created scent object against a database to determine whether the combination of chemicals used for the scent may be deleterious to health, deleterious to people with certain conditions such as specific allergies, or to determine if the combination of chemicals presents any other hazard, such as to the scent emitting unit or other equipment (for example, combining a volatile acid with a volatile amino compound may produce a solid precipitate, clogging the scent emission unit). In the event a problem or hazard is detected, the user can be warned appropriately, the scent object can be rejected from the registry or listed as inoperative, or other remedial action can be taken.

Furthermore, advertisers or content developers wanting to use a proprietary smell can do so by custom tailoring the electronic description of their unique scent. They may also develop the proprietary scent as a single output odorant substance for emission by the emitter. For example, if a particular perfume maker wishes to advertise with its own scent instead of a generic perfume scent, the maker could provide its own electronic description to be used with

the electronic scent technology described and disclosed herein. Alternatively, to provide users with a true representation of the perfume, the perfume maker could provide a sample or distillate of the perfume to incorporate into the emitter subsystem. If a cartridge device is used for the emitter system, the advertiser can purchase the right to have a particular scent placed in the cartridge. This effectively divides the space available for storing odorant substances in the emitter system into "real estate" which is then purchased by one or more companies who wish to make their particular scents available for emission by the emission system. In this embodiment of the invention, the company purchasing the right to have their scents placed in the emission system will typically supply the actual scent to be placed in the emitter, such as the sample or distillate provided by the perfume manufacturer in the above example.

An additional application of scent technology, which is particularly suitable for advertising, marketing, and promotion, is to associate a scent with an entity or action that normally does not have a scent associated with it, or associating a scent with an entity or action other than the scent commonly associated with that entity or action. One example of this application is to associate a scent with a company name or company logo; e.g., a brokerage firm advertising on the Internet using a banner ad may wish to have the scent of freshly-printed money emitted when a user views or clicks on the ad; a residential developer may wish to have scents associated with the home (e.g., freshly baked apple pie, baby powder, or the like) emitted when a user visits its Web site. An example of a scent not normally associated with an entity or action is the scent of damp earth associated with a sports utility vehicle.

User-created scents and other scents can be listed on a central Web site for sharing with other users, or can be placed in a protected area of a Web site for use only by a limited subset of users, perhaps only one user. A Web site can have a list of freely available scents, and another list of limited access scents, which can be accessed by paying a license fee, subscription fee, or applying for access. Access can then be granted to sample the scent once, a limited number of times, or freely; or permission can be granted to link the scent to the subscriber's Web site or download the scent object, or present the scent on another device (e.g., a standalone scent kiosk).

Items and services which utilize scent include, but are not limited to, perfumes, colognes, perfumed articles, candles, incense, sachets, air fresheners, or in cosmetics; in lotions, ointments, and creams for application to the skin; in hair care products such as shampoos, conditioners, hair sprays, hair gels, hair mousses, soaps, etc. in personal care products such as soaps, deodorants, antiperspirants, shaving creams, aftershave lotions,

mouthwashes, toothpastes, respiratory care products and other medicines (e.g., vapor rubs, cough drops, etc.). One of skill in the art will recognize that for any product utilizing a scent, the scent chemical must be safe for the application used; for example, scents used in lotions or deodorants must be safe for application to the skin; scents used with mouthwashes or toothpastes must be safe for contact with the human oral cavity.

Other items which utilize scent include, but are not limited to, solid or liquid anionic, nonionic, cationic or zwitterionic detergents; fabric softener articles; dryer-added fabric softener articles; fabric softener compositions; cosmetic powders; hair preparations; perfumed polymers and the like; washes and rinses for fruits and vegetables (using scents that are safe for human consumption on foods); laundry and cleaning products, such as dish detergents, water softeners, fabric conditioners, hard surface cleaners (e.g. floor cleaners, floor waxes, tile cleaners, deodorizers and disinfectants, etc.) and laundry detergents, fabric softeners, and bleaches (when the bleaching component is compatible with the scent components) Additional items which utilize scent include, but are not limited to, hygienic paper products such as diapers, disposable undergarments, feminine protection products such as sanitary napkins and tampons, facial and bathroom tissues and towels, and wipes. Other paper products include, but are not limited to, stationery, postcards, stamps, decals, stickers, business cards, greeting cards, posters, calendars, photographs, household paper towels, paper cups, plates, and other eating utensils (using scents safe for contact with food). Scents can also be incorporated into foods and beverages, including, but not limited to, wines, beers, candy, soda, mineral water or other water beverages, chewing gum, and processed foods, where such scents are safe for consumption with the food and/or beverage. Scents can also be used in conjunction with electronic systems. Examples include, but are not limited to, transmission of scents via email ("scented email"); scents coordinated or associated with system operations (e.g., startup scents, shutdown scents, error scents, and system clock driven scent alarms; microprocessor-driven home and/or car scents, where emission of scents is controlled by a microchip or a computer driven home and/or car scent system(s), with emission of scent based on time and/or location, or on demand via voice or traditional interfaces.

While the invention described herein has been referred to primarily in terms of the human olfactory system, the invention is not limited to the human sense of smell, or even mammalian olfactory systems. The invention can be used with any animal which responds to odorant substances. For example, a human operator can generate or edit scents for pet food, and a panel of dogs, cats, or other household pets can be screened to test the appeal of a particular generated scent. Scents can be tested on mice, rats, guinea pigs, and other rodents

and animals, for testing purposes, research purposes, or rodent control purposes. Insects are known to be highly responsive to odorant substances, and the invention can be applied for purposes of insect control (such as repelling unwanted insects from a house or outdoor location, or attracting beneficial insects to agricultural sites). Thus, while the invention has been primarily described in terms of the human olfactory system and human olfactory receptors, it can applied to any animal responsive to scents and odors.

Scent selection by users can also provide useful information for other commercial purposes. As mentioned herein, the invention embraces the customization of products such as computer games, audio recordings (such as compact disks) or video recordings (such as DVD's) or other products with scent tracks downloaded from a database. Tabulation of the usage of scents in each application can provide vendors with useful data, particularly the use of scents associated with particular features of the product. For example, statistics can be compiled regarding the frequency with which particular scents are added to characters in a computer game, the frequency with which particular scents are added to certain types of music, or the frequency with which particular scents are added to categories of greeting cards. A vendor can then examine such data for useful information regarding the preferences of scents associated with a particular product. Users adding scents to greeting cards, for example, may prefer specific floral type smells for Valentine's Day-related cards over other types of smells, e.g., lavender may be preferred over rose by a 2-to-1 ratio, or all floral scents may be preferred over all other scents by a 4-to-1 ratio, or all floral scents may be preferred over food-related scents by a 100-to-1 ratio, when applied to the Valentine's Day greeting cards. Vendors can use such information to customize products (such as Valentine's Day greeting cards sold in ordinary stores) with the preferred scents, or for other marketing or commercial purposes. Similarly, if purchasers of classical music prefer floral type smells when customizing audio recordings with scent tracks, while purchasers of rock music may choose scents associated with large crowds, such as sweat or smoke. A vendor can then use this information to customize products (such as compact disks or other audio recordings sold in ordinary stores, or on-line via non-scent-enabled computers) with the preferred scents, or for other marketing or commercial products. Similarly, tabulation of data regarding scent tracks added to movies by various users can provide information to the vendor about preferred scents associated with particular scenes.

The data regarding scent selection can be sorted using various demographic criteria, e.g., age, sex, income level, by preferred hobbies or activities, by region or locality, or by other demographic categories, in order to target particular markets. For example, a vendor may

observe that purchasers of greeting cards in Louisiana prefer magnolia-scented cards, while purchasers of greeting cards in Maine prefer rose-scented cards, and can target their distribution (whether on-line or in traditional sales outlets) accordingly.

In another embodiment of the invention, scent vendors can place their own scent tracks on products such as games, musical recordings, movies, etc., as a "scent product placement." This is especially useful when the scent is associated with a particular feature of the product, such as a character in a video game or movie, the climax of a musical performance, or an exciting scene in a video game. A maker of perfumes may wish to have their perfume associated with a particular character in a video game or other video display; e.g., a movie featuring an attractive heroine may contain a scent track specifying the delivery of a proprietary perfume scent when the heroine is on screen. Likewise, a movie with scenes where food or beverages are consumed may contain a scent track specifying the delivery of the scent of a particular brand of food or beverage when those scenes are displayed; a movie with scenes of people on a beach may contain a scent track specifying the delivery of the scent of a particular brand of suntan lotion when those scenes are displayed. In an analogous manner, scent tracks can be placed on music recordings such as compact disks, DVD's, computer games, video games, TV and radio broadcasts, and other multimedia presentations. Examples of features with which the scents may be associated include, but are not limited to, a character in a game or movie; an action in a game or movie; an environment in a game or movie; a scene in a game or movie; an object in a game or movie; a period of time in a game, movie, or musical recording; a level in a game; a phase or stage of a game, movie, or musical recording; a score kept in a game; progress made in a game or movie; a particular sound in a musical recording; a particular musical instrument in a musical recording; or a particular voice or singer in a musical recording.

Fig. 14 is a pictorial representation of some of the applications of the electronic scent concept of the present invention.

The process and system of the present invention has been described above in terms of functional modules in block diagram format. It is understood that unless otherwise stated to the contrary herein, one or more functions may be integrated in a single physical device or a software module in a software product, or a function may be implemented in separate physical devices or software modules, without departing from the scope and spirit of the present

invention. For example, referring to Fig. 1, the recording subsystem 12 or components thereof may be embodied in a personal computer, and the emission subsystem 40 may be embodied in a television console or a standalone scent player, or both the recording and emission subsystems may be integrated in a single device. Other variations may be possible depending on the specific application.

The following examples are provided to illustrate the invention, and should not be construed as limiting the invention in any manner.

Example 1

User-Directed Scent Creation and Customization

The invention provides users with the ability to custom-tailor scents. This is done by presenting the user with a series of components, which can be mixed and matched at the user's direction, and then sampled by emitting the combination of components specified by the user. Table 1 shows a list of odorants which can be loaded in the scent emitting unit. The user is presented with a computer menu (such as a Web page display) which provides a categorization of odorants or scent chemicals, either by the full (chemical) name of the scent; by one or more of the flavors (odors) associated with that scent; by keywords (woody, citrus, etc.) associated with the scent; by category (e.g., fruits, flowers, perfumes); by preferred application of the scent (e.g., detergent, aftershave, food additive); by any other characteristic of the odorant; or by any combination of those terms. The preferred arrangements are to provide the flavor or flavors associated with the odorant or the keyword associated with the scent, with the option of viewing the corresponding chemical name and other properties if desired. The computer menu allows selection of the intensity of the scent; this can be done in terms of relative scent intensity versus a standard odorant (see Example 2), by absolute concentration of the scent (e.g., in parts per million), or on an arbitrary scale. The user selects particular components to blend. This could be implemented on a Web page by, for example, selecting a check box to indicate the presence of that component in the mixture. Upon selection of a component, the user is then prompted to enter the intensity of that component. This can be implemented on a Web page by, for example, using a sliding control or entering a numerical value (between specified limits) or choosing a radio button corresponding to an intensity in order to indicate the intensity of the selected component. A default intensity can be provided if desired. This procedure provides the user with a "scent palette," that is, a scent mixing interface which can

be manipulated to create new scents, analogous to the creation of new colors with a color palette. In the terms of the embodiment of a scent object previously described, the user is editing the Palette Vector of the scent object.

When all desired components have been selected and their intensities entered, the user indicates that the mixture is to be emitted by the scent emitting unit. This can be implemented on a Web page by simply having an "EMIT" button or a "SMELL" button. The computer sends information regarding the components and intensities chosen to the scent emitting unit. The scent emitting unit then mixes the selected odorants in the specified intensities and emits the mixture. The user can then smell the mixture, and decide whether particular odorants should be added or deleted, or if odorant intensities should be adjusted. Thus, for example, a user trying to create a custom chocolate scent can select the flavor "cocoa" (represented by, e.g., 2, 5-dimethylpyrazine) which has in an arbitrary concentration of "5." Upon sampling the emitted odor, the user may decide to blend a "hazelnut" scent (represented by, e.g., 5-methyl-2-hepten-4-one) the "cocoa" scent, at an arbitrary concentration of "3." The user proceeds in this fashion, adding odorants, adjusting their intensities, and sampling the results, until the desired combination is arrived at. A scent object is then generated which corresponds to the desired combination.

Alternatively, instead of starting the scent design de novo, the user can select a pre-mixed scent, which they can then proceed to customize. The user can be presented with a list of scents, and can select a scent to customize. This can be implemented on a Web page using a drop-down menu. For example, a chemical recipe used to simulate the scent "cinnamon" is

Cinnamon Flavor (1)
alpha-Pinene 0.2%
1,8-Cineole 1.65%
p-Cymene 0.55%
Linalool 2.3%
Caryophyllene 1.35%
alpha-Humulene 0.2%
alpha-Terpineol 0.4%
Cuminaldehyde 0.25%
Cinnamaldehyde 74%
Cinnamyl acetate 5.1%
Eugenol 8.8%

Benzyl benzoate 1.0%

Camphor 2.1%

Acetoeugenol 2.1%

Upon selecting the Cinnamon Flavor (1) from the menu of pre-mixed scents, the "recipe" comprising the chemical odorant components of the scent will be presented to the user. The user can de-select an odorant, add odorants not originally present in the mixture, or adjust the intensity of one or more odorants. The user can sample the scent at each step until a desired scent is arrived at. In some instances, the user may simply sample the initial scent, and then select the initial scent recipe as the desired scent recipe; the appropriate scent object can be generated from the scent recipe at that point, or can be previously stored. The "recipe" for the scent can also be presented to the user in descriptive terms, by presenting a list of odorant descriptors for each chemical components, or selected chemical components, instead of or in addition to the chemical names of the odorant components; for example, the Cinnamon Flavor (1) recipe above may list "spicy/clove-like" in place of eugenol, and have a numerical rating (on a scale of 1 to 10, or 1 to 100) or a descriptive rating (e.g., overpowering/strong/moderate/weak/very weak). The user could then adjust the numerical or descriptive rating to enhance or attenuate that component of the scent.

Once the desired combination is arrived at, the user can store the scent object corresponding to the custom scent in memory, for example, as part of a scent database. The custom scent can also be communicated or transmitted, for example, by email or by placing a scent link on a Web site, so that a remote user can sample the scent. The user can communicate or transmit the scent to another party, such as a commercial vendor, which can then blend the custom scent into products of the user's choice. For example, the user may wish to have a scent blended into a shampoo, a soap, candles, or an air freshener, or imprinted on stationery or postcards. Alternatively, the scent can be supplied in a vehicle suitable for application to the skin, in order to create a customized perfume created according to the user's specfication.

Pre-mixed recipes can also be used for "scent polling." A scent, or a variety of scents, are presented as standard recipes (the ability to alter the combinations typically is not provided in the scent polling application). The user can sample each premixed scent using the scent emitter, and rank or "vote" for each scent in order of preference, or by assigning a desirable/undesirable value to the scent (e.g., on a scale of -10 for undesirable to +10 for desirable), or by other rating systems. This information can then be communicated or

transmitted to another party, such as a consumer products vendor. The vendor can analyze the results of the scent polling to determine which scents would be most appropriate to mix into various products. For example, a Web page can be entitled "How would you like your shampoo to smell?" with various scent recipes available to sample. Users could vote on each scent, and the vendor could then use the information accordingly in formulation of their products in order to enhance the likelihood of commercial success.

Table 1

Full Name	Flav1	Flav2	Flav3	Flav4	Flav5	Flav6	Flav7	Flav8
Allyl isothiocyanate	pungent	strong						
Allyl isovalerate	winey	fruity	fermented	apple				
Allyl mercaptan	garlic	diffusive	onion	sweet	leek			
Allyl methyl disulfide	asparagus	sweet						
Allyl methyl trisulfide	garlic							
Allyl nonanoate	sweet	waxy	fruity	winey	fatty	oily	pineapple	
Methylpyrazine	nutty	cocoa	green	roasted	chocolate	meaty	brown	musty
2,3-Dimethyl-pyrazine	green	nutty	coffee	"peanut butter"	walnut	cocoa	caramel	meat
2,5-Dimethyl-pyrazine	cocoa	"roasted nut"	"roast beef"	woody	grassy	"potato chip"	medicinal	"grilled chicken"
2,6-Dimethyl-pyrazine	cocoa	"roasted nut"	"roast beef"	coffee				
Trimethyl-pyrazine	"roasted nut"	cocoa	"baked potato"	peanut				
Tetramethyl-pyrazine	chocolate	musty	sweet	coffee	cocoa	lard	"fermented soybean"	burnt
2,6-Dimethyl-3-ethylpyrazine	coffee	"roasted nut"	cocoa	"potato chip"	"burnt almond"	chocolate	woody	
5-Ethyl-2-methylpyridine	fatty	green	nutty					
2-trans-4-trans-Decadienal	"deep fat fried"	fatty	citrus	powerful	chicken			
gamma-Decalactone	fruity	peach	fatty	sweet	creamy			
delta-Decalactone	creamy	coconut	peach	buttery				
epison-Decalactone	coconut	caramel	nutty					
Decanal	sweet	penetrating	waxy	floral	citrus			
delta-Damascone	rosy	fruity	"black currant"	sweet	intense			
Decanoic acid	fatty	rancid	citrus					

SUBSTITUTE SHEET (RULE 26)

2-trans-6-cis-Dodecadienal	waxy	fatty	citrus	melon	green	aldehydic	spice	
2-trans-4-trans-Dodecadienal	fatty	citrus						
gamma-Dodecalactone	fruity	musk						
delta-Dodecalactone	fruity	peach	buttery	coconut	fatty	waxy	dairy	milky
epison-Dodecalactone	peach	milky	fatty					
Ethyl alcohol	alcoholic	ethereal						
2-Acetyl-1-Ethylpyrrole	floral	cherry	weak					
Ethyl decanoate	brandy	oily	fruity	grape				
Methyl sulfide	raddish	diffusive	repulsive	"asparagus taste"	sharp	cabbage	green	
Tetramethyl ethylcyclohexenone	caramel	warm	fruity	spicy				
2-(p-Tolyl)-propion-aldehyde	sweet	intense	refreshing	peppermint				
Dehydro-menthofuro-lactone	coumarin	hay						
4-Ethyl-benzaldehyde	"bitter almond"	cherry						
Ethyl methyl-p-tolylglycidate	"cooked fruit"	dry	fruity					
5-Hydroxy-8-undecenoic acid, *delta-lactone*	floral	milky	buttery	creamy	sweet	fruity	milky	dairy
5-Isopropenyl-2-methyl-2-vinyl-tetrahydrofuran	herbaceous	piney	fresh	lime	pungent			
5-Methyl-2-hepten-4-one	hazelnut	metallic	buttery	nutty	pungent	roasted	ketonic	coffee
3-Methyl-1-pentanol	winey	fruity	green	pungent				
Dihydro-jasmone	floral	jasmine	celery	fruity	herbaceous	spicy	sweet	woody
Mintlactone	sweet	coumarin	creamy	coconut				
Thiomenthone	cassis	blackcurrant	herbal	fruity				
Thujone	herbal	medicinal	phenolic	artemisia				

SUBSTITUTE SHEET (RULE 26)

7-Acetyl-1,1,3,4,4,6-hexamethyl-tetralin	strong	persistent	nitro-free	musk				
5-Acetyl-1,1,2,6-tetramethyl-3-isopropyl-indane	nitro-free	aromatic	musk	sweet	floral	note		
Acetic acid, alpha-(trichloromethyl) benzyl ester	heavy	floral	powdery	rose				
Tridec-2-enenitrile	fatty	aldehydic	<0.01% ozone					
2,4-Dimethyl-3-cyclohexene-1-carboxaldehyde	herbal	harsh	green					
Vanillonitrile	mild	aromatic	sweet	vanilla				

Other scent recipes which can be edited on-line include the following:

Cheese Flavor (I):

Acetic acid 3.14%; Propanoic acid 4.74%; Hexanoic acid 0.34%; Decanoic acid 2.04%; Pyrrolidine 0.07%; Butyric acid 1.35%; Palmitic acid 0.36%; Methyl sulfide 0.15%; Diacetyl 2.41%; Ethyl vanillin 10.15%; Lactic acid 1.01%; 2,4-Decadienal 1.41%; Acetoin 0.74%; Propylene glycol 72.09%;

Cheese Flavor (II):

Butyl butyryl lactate 2 oz; Isovaleric acid 2 oz; Ethyl butyrate 2 oz; Hexanoic acid 1 oz; Methyl n-Amyl ketone 1 oz; Alcohol (95%) 16 oz; Propyleneglycol 102 oz;

Coconut Flavor (I):

Nonalactone 3%; Aldehyde C-1 82%; Vanillin 1.5%; Maltol 1%; Piperona l0.2%; Lauric acid 0.2%; Cyclotene 0.2%; Ethyl Alcohol 45%; Propylene glycol 46.9%;

Coconut Flavor (II):

Vanillin 6%; Ethyl vanillin 6%; Alcohol (95%) 48.25%; 1-Octanol 0.75%; Hexyl alcohol 3%; Ethyl oenanthate 3%; Aldehyde C-18 33%;

Cinnamon Flavor (I):

alpha-Pinene 0.2%; 1,8-Cineole 1.65%; p-Cymene 0.55%; Linalool 2.3%; Caryophyllene 1.35%; alpha-Humulene 0.2%; alpha-Terpineol 0.4%; Cuminaldehyde 0.25%; Cinnamaldehyde 74%; Cinnamyl acetate 5.1%; Eugenol 8.8%; Benzyl benzoate 1.0%; Camphor 2.1%; Acetoeugenol 2.1%;

Cinnamon Flavor (II):

1,8-Cineole 4.7%; Cinnamaldehyde 62.3%; Linalool 6.3%; Benzaldehyde 1.9%; alpha-Terpineol 1.6%; beta-Caryophyllene 2.8%; Cinnamyl acetate 7.8%; Eugenol 6.3%; Cinnamon bark oil 6.3%;

Jasmin Base (I):

Benzyl acetate 65%; Benzyl propionate 5.0%; Benzyl butyrate 2.0%; Linalool 6.0%; Linalyl acetate 2.0%; Hydroxycitronellal 6.0%; Phenethyl alcohol 7.0%; p-Cresyl phenylacetate 0.5%; Isoamyl valerate 0.3%; Ethyl anisate 0.3%; Eugenol 0.2%; Bitter orange oil 0.2%; Amyl cinnamic aldehyde 5.5%;

and Jasmin Base (II):

Benzyl alcohol 11%; Linalyl acetate ex bois de rose 11%; Benzyl acetate 19%; Benzyl benzoate 6%; Ethyl phthalate 3%; Jasmin chassis absolute, ex benzene 6%; Laurine 10%; Phenethyl alcohol 8%; Linalool 9%; Bergamot oil, sesquiterpeneless 4%; Peru balsam 5%; Ylang-ylang oil 3%; Benzyl salicylate 5%.

Example 2

Describing Odors in Terms of Perceptive Primary, Quasi-Primary, and Complex Scent Components

Reference is made herein to the publication designated E544-99 by the American Society for Testing and Materials (Annual Book of ASTM Standards, available from ASTM, 100 Barr Harbor Dr., West Conshohocken, Pennsylvania, USA) entitled "Standard Practices for Referencing Suprathreshold Odor Intensity" (1999), referred to as ASTM E544-99; this publication is hereby incorporated by reference herein in its entirety. Other ASTM publications of relevance to scent detection include ASTM E1593-94 and ASTM E679-91; those publications are hereby incorporated by reference herein in their entirety. The procedure outlined in ASTM E544-99 is a method to reference the intensity of an odor in the suprathreshold region. The procedure relates the intensity of the odor to the odor intensity of a known concentration of n-butanol (the reference odor). This can be used to establish a scale of odor intensities relative to the single reference odor.

The scale thus generated provides the magnitude of a perceptive primary scent component, perceptive quasi-primary scent component, or perceptive complex scent component. While the odor of n-butanol is non-linear with concentration, and the odor of an arbitrary odor is also likely to be non-linear with concentration, generating relative scales of odor intensities allows linear measures of subjective odor intensities to be mapped to the actual concentrations. That is, if a given concentration of odor X is judged to be equal in intensity to 1000 ppm of n-butanol, while a given concentration of odor Y is judged to be equal in intensity to 200 ppm of n-butanol, the perceived odor intensity ratio is given by $X/Y = (1000/200)^{0.66} = 2.9$, as it is know that on average, the odor intensity of n-butanol changes in proportion to its concentration in air raised to the 0.66 power. Thus, the magnitude of odor intensity at those two concentrations is 2.9. Measurement of the odor intensity is performed at a wide range of concentrations of odors X and Y, typically, at the concentrations which will be used in practice. This establishes the relative intensities at the concentrations of interest, and allows combination and blending of the appropriate amount of odorant in order to re-create a scent or component of scent as desired.

To implement the method, a panel of ten subjects is selected according to ASTM E544-99 Section 5.5. A selection of odorant molecules is selected which provides a range of

sensory stimulation; see, e.g, Table 1, Example 1. The subjects rate the odorants according to intensity versus n-butanol. Equipment and protocols for this procedure are described in ASTM E544-99.

Once the intensity of the various odorants has been established, the intensity scale can be used in a variety of applications. One such application is in the creation of custom scents in Example 1 where the intensity scale can be used to calibrate the amount of each odorant to be emitted. That is, the known concentrations of a scent on the intensity scale--e.g., the weakest and strongest concentrations--can be used as, for example, the endpoints on the adjustable scale specified by the user. The absolute amount of each odorant to be added will be heavily odorant-dependent, and by using the intensity scale, users can be prevented from entering, e.g., 1200 ppm for an odorant which is overpowering at 100 ppm, or entering 50 ppm for an odorant which can barely be sensed at 500 ppm.

The intensity scale can also be used to generate descriptions of unknown scents in terms of perceptive primary, quasi-primary, and complex scent components. An unknown scent can be provided to the scent panel, which compares the unknown with each of the odorants which have been previously ranked in intensity versus n-butanol. The panel then rates the unknown scent versus each odorant in terms of similarity. Thus, for example, a panel member, presented with the sample scent of hot chocolate, will rank a "chocolate" odorant (e.g., tetramethylpyrazine)as more similar than an "herbal" odorant (e.g., 2,4-dimethyl-3-cyclohexene-1-carboxaldehyde). (See Table 1, Example 1). For each odorant, then, an intensity level is generated in comparison to the unknown scent. In the illustration using only two odorants, "chocolate" and "herbal," as the perceptive primary scent components, the resulting vector for the ordered set (chocolate, herbal), would appear as (1, 0). Most applications will utilize many more odorants, with correspondingly larger vectors representing the scents.

This method can also be used without first ranking the odorants versus n-butanol, although relative scent intensities of the odorants with respect to each other must then be determined at the same time as the relative intensities of the odorants with respect to the unknown sample scent.

The method described provides a means of generating a description of a scent in terms of a set of odorants. In terms of the embodiment of a scent object described previously, a Hedonic Vector is generated, where the reference set of the Hedonic Vector is the set of odorants against which the sample scent is compared. The method also provides a means of generating the scent from the set of odorants used for comparison. Again in terms of the

previously described embodiment of a scent object, a Palette Vector is generated, where the chemicals used in the reference set are substituted for each descriptive category used to describe the scent.

Example 3

Computer Simulations of Odor Similarity

The methodology used herein is described in A. N. Jain. Morphological Similarity: A 3D Molecular Similarity Method Correlated with Protein-Ligand Recognition. *Journal of Computer-Aided Molecular Design.* 14: 199-213, 2000; this publication is hereby incorporated by reference herein in its entirety.

Computer simulations were performed in order to demonstrate that morphological comparisons can be used to compare molecular structures of odorants, and that molecular structures with similar morphology display similar odor characteristics. Essentially, computer-generated comparisions of of odorant molecules, by using Euclidean distance calculations between vectors corresponding to reference set similarity calculations, provides a means of categorizing molecules of known structure, but unknown scent.

Eleven odor categories were designated. These were, in alphabetical order:

banana
citrus
floral
herbaceous
meaty
medicinal
minty
oily
onion
rosy
vegetable

In each category, six distinct chemical structures taken from various flavor/odor databases were chosen. Each chemical structure was accompanied by data showing the category label to be the "primary" odor.

3-D molecular models of these chemical structures were obtained using the MM2 force field with a commercial molecular modelling application, Molecular Modeling Pro v3.24, from ChemSW Inc. (Fairfield, CA). Default settings were used. The resulting energy minimized models were compared for "morphological similarity" with SurFlex-Sim v1.0 from BioPharmics (San Mateo, CA), a 3-D chemical conformational shape comparison program. Default settings were used in this program as well.

Next, the morphological similarity values (or "distances") were compared in two ways: "like" odors (those within a odor category) were contrasted with "unlike" odors (those from different odor categories). The distance value used was Euclidean distance calculation among vectors of the respective sets.

The two sets of values were collected in separate histograms and a gaussian curve-smoothing function was applied to the histograms to obtain the different peaks seen in Figs 15-24.

Protocol for simulation

Database entry flavor descriptors (FD) were correlated with (reference set) vector data from a "whole database" perspective. These correlations were performed in order to determine the patterns generated by comparing entries "like" a given entry with respect to flavor descriptors "unlike" a given entry. That is, pairwise comparisons of FDs within a class were performed, as well as pairwise comparisons of FDs across classes. Each computational run used a large subset of the database Euclidean distance calculations in order to obtain on the order of 2000 pairwise Euclidean comparisons, which were then graphed. (See Figures 15-24.)

Each computational run used a standard sized set of randomly chosen flavor-database entries which met the four criteria as outlined below:

1. Entries rich in flavor descriptors (FDs) were chosen at random from the database, meaning that all possessed at least six FDs. These entry shall be referred to as the SEEDs below.

2. Further DB entries were chosen which had above a threshold number of like entries to the SEEDs as defined in 3) below. In addition, for each SEED, the ratio of entries "like

SEED" to those "unlike SEED" had to be above another threshold, to eliminate as many potential "outliers" from the survey as possible.

3. Entries were judged to be "like" the SEED if, when entry A was compared with SEED, entry A shared at least half of SEED's FD's. The reverse comparison of SEED to A was ignored.

4. Entries were judged "unlike" only if A did not share any of SEED's flavor descriptors at all. Again, the reverse comparison of SEED to A was ignored.

Lists of entries "like" and "unlike" each SEED were compared to within lists. Four different sets of vectors were used to calculate the Euclidean distance within each group:

a. The original 20-component vector set, which was "most orthogonal" within a small subset of the database.

b. A second 20-component vector set, derived by choosing a "20 most orthogonal" set from the entire database.

c. A third 12-component vector set, comprised of structures that have a "strong smell", chosen from the set of "128 most orthogonal" in the entire database.

d. A vector set made by combining set b) and set c).

All the vector sets generated in the computational section were found to be effective to some degree in separating "like" from "unlike." Set b) (see Figs. 17 and 18) appeared most effective. Less effective was set a) [See Figs 1&2], with somewhat better separation seen with sets b) and c). No substantial improvement was seen by lengthening the vectors in set d). (See Figures 21 and 22). Indeed, vector sets made of shorter vectors (set c), 12 components) were almost as effective at separating "like" from "unlike" as long vectors.

The vector set with the most effectiveness, set b) (Figs. 17 and 18) contains a number of "odorless" molecules. By including "smellier" molecules as vector components, it is possible that a greater separation of smells can be achieved. Figs. 21 and 22 exemplify one attempt at this approach, although separation is not appreciably greater in those simulations.

In order to take into account the effect of which particular entries are run (in what order), several runs using each vector set were attempted. Except for minor variations in histogram modalities (see Figures 23 and 24), gross graph characteristics stayed relatively constant.

The applications of this morphological similarity analysis are myriad. Given that chemical analysis of gaseous mixtures can often be automated, an unknown sample scent can

be subjected to chemical analysis by, for example, GC/MS analysis, to identify each chemical component of the sample scent. Re-creation of the sample scent would then be possible by re-combining each of the component molecules in their appropriate concentrations. In practice, however, the scent emitter module will have a limited set of odorants to use for scent re-creation. A morphological comparison of the molecules identified in the sample scent versus each of the available odorants can determine which odorant or odorants available in the emitter module are most similar to each component of the sample scent. These odorants can then be combined to provide a simulation of the sample scent. In terms of the embodiment of the scent object previously described, this method generates a Structural Vector component of the scent object. If desired, the user can use the method outlined in Example 1 to refine and adjust the scent as desired.

It is appreciated that detailed discussion of the actual implementation of each module is not necessary for an enabling understanding of the invention. The actual implementation is well within the routine skill of a programmer and system engineer, given the disclosure herein of the system attributes, functionality and inter-relationship of the various functional modules in the system. A person skilled in the art, applying ordinary skill can practice the present invention without undue experimentation.

All references, including patents, patent publications, patent applications, scientific articles and references, technical articles and references, and all other publications mentioned herein, are hereby incorporated by reference herein in their entirety.

While the invention has been described with respect to the described embodiments in accordance therewith, it will be apparent to those skilled in the art that various modifications and improvements may be made without departing from the scope and spirit of the invention. Accordingly, it is to be understood that the invention is not to be limited by the specific illustrated embodiments, but only by the scope of the appended claims.

APPENDIX A

```
// Implementation of: public class ChemSimilartyValue

#include "ChemSimilartyValue.h"

// ---------------------------------------------------------
ChemSimilartyValue::ChemSimilartyValue ()
{
}

// ---------------------------------------------------------
ChemSimilartyValue::ChemSimilartyValue (const ChemSimilartyValue&)
{
}

// ---------------------------------------------------------
ChemSimilartyValue::~ChemSimilartyValue ()
{
}

// ---------------------------------------------------------
ChemSimilartyValue& ChemSimilartyValue::operator = (const
ChemSimilartyValue &arg)
{
      return *this;
}
```

```
#ifndef _ChemSimilartyValue_H_
#define _ChemSimilartyValue_H_

#define exception class
// public class: ChemSimilartyValue
class ChemSimilartyValue {
public:
      ChemSimilartyValue ();
      ChemSimilartyValue (const ChemSimilartyValue&);
      virtual ~ChemSimilartyValue ();
      ChemSimilartyValue& operator = (const ChemSimilartyValue &arg);

// Attribute Get/Set Methods:
      const integer&    GetRefCompoundID () const     {return
RefCompoundID;}
      void  SetRefCompoundID (const integer &val)     {RefCompoundID =
val;}

// Attributes:
private:
      integer     RefCompoundID;
      double      SimilarityScore;
};
#endif
```

```
public class ChemSimilartyValue {
      public ChemSimilartyValue () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:
      public integer GetRefCompoundID ()  {return RefCompoundID;}
      public void SetRefCompoundID (integer val)      {RefCompoundID =
val;}

// Attributes:
      private integer RefCompoundID;
      private double SimilarityScore;
}
```

```
// Implementation of: public class Compound

#include "Compound.h"

// ---------------------------------------------------------------
Compound::Compound ()
{
}

// ---------------------------------------------------------------
Compound::Compound (const Compound&)
{
}

// ---------------------------------------------------------------
Compound::~Compound ()
{
}

// ---------------------------------------------------------------
Compound& Compound::operator = (const Compound &arg)
{
      return *this;
}
```

```
#ifndef _Compound_H_
#define _Compound_H_

#define exception class
#include "StructuralVector.h"
#include "PolymerReactivityVector.h"
#include "ReceptorScreeningVector.h"
#include "ReceptorDockingVector.h"
#include "StructuralVector.h"

// public class: Compound
class Compound {
public:
      Compound ();
      Compound (const Compound&);
      virtual ~Compound ();
      Compound& operator = (const Compound &arg);

// Attribute Get/Set Methods:
      const integer&     GetCompoundID () const  {return CompoundID;}
      void  SetCompoundID (const integer &val)  {CompoundID = val;}
      const String&      GetCommonName () const  {return CommonName;}
      void  SetCommonName (const String &val)   {CommonName = val;}
      const String[]&    GetStructure () const   {return Structure;}
      void  SetStructure (const String[] &val)  {Structure = val;}
      const String&      GetSmiles () const      {return Smiles;}
      void  SetSmiles (const String &val) {Smiles = val;}
      const double&      GetVaporPressure () const     {return
VaporPressure;}
      void  SetVaporPressure (const double &val)      {VaporPressure =
val;}
      const double&      GetVaporTemperature () const  {return
VaporTemperature;}
      void  SetVaporTemperature (const double &val)   {VaporTemperature =
val;}
      const double&      GetFlashPoint () const  {return FlashPoint;}
      void  SetFlashPoint (const double &val)   {FlashPoint = val;}
      const double&      GetViscocityRating () const   {return
ViscocityRating;}
      void  SetViscocityRating (const double &val)    {ViscocityRating =
val;}
      const double&      GetOxidationRating () const   {return
OxidationRating;}
      void  SetOxidationRating (const double &val)    {OxidationRating =
val;}
      const double&      GetSolubility () const  {return Solubility;}
      void  SetSolubility (const double &val)   {Solubility = val;}

// Attributes:
private:
      integer     CompoundID;
      String      CommonName;
      String[]    Structure;
      String      Smiles;
      double      VaporPressure;
      double      VaporTemperature;
      double      FlashPoint;
      double      ViscocityRating;
      double      OxidationRating;
```

```
        double      Solubility;

// Relationships:
public:
        PolymerReactivityVector *unnamed;
        ReceptorScreeningVector *unnamed;
        ReceptorDockingVector   *unnamed;
};
#endif
```

```
public class Compound {
      public Compound () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:
      public integer GetCompoundID ()      {return CompoundID;}
      public void SetCompoundID (integer val)   {CompoundID = val;}
      public String GetCommonName ()       {return CommonName;}
      public void SetCommonName (String val)    {CommonName = val;}
      public String[] GetStructure ()      {return Structure;}
      public void SetStructure (String[] val)   {Structure = val;}
      public String GetSmiles ()     {return Smiles;}
      public void SetSmiles (String val)  {Smiles = val;}
      public double GetVaporPressure ()   {return VaporPressure;}
      public void SetVaporPressure (double val) {VaporPressure = val;}
      public double GetVaporTemperature ()      {return VaporTemperature;}
      public void SetVaporTemperature (double val)    {VaporTemperature =
val;}
      public double GetFlashPoint ()       {return FlashPoint;}
      public void SetFlashPoint (double val)    {FlashPoint = val;}
      public double GetViscocityRating () {return ViscocityRating;}
      public void SetViscocityRating (double val)     {ViscocityRating =
val;}
      public double GetOxidationRating () {return OxidationRating;}
      public void SetOxidationRating (double val)     {OxidationRating =
val;}
      public double GetSolubility ()       {return Solubility;}
      public void SetSolubility (double val)    {Solubility = val;}

// Attributes:
      private integer CompoundID;
      private String CommonName;
      private String[] Structure;
      private String Smiles = String;
      private double VaporPressure;
      private double VaporTemperature;
      private double FlashPoint;
      private double ViscocityRating;
      private double OxidationRating;
      private double Solubility;

// Relationships:
      public PolymerReactivityVector unnamed;
      public ReceptorScreeningVector unnamed;
      public ReceptorDockingVector unnamed;
}
```

```
// Implementation of: public class HedonicValue

#include "HedonicValue.h"

// ---------------------------------------------------------
HedonicValue::HedonicValue ()
{
}

// ---------------------------------------------------------
HedonicValue::HedonicValue (const HedonicValue&)
{
}

// ---------------------------------------------------------
HedonicValue::~HedonicValue ()
{
}

// ---------------------------------------------------------
HedonicValue& HedonicValue::operator = (const HedonicValue &arg)
{
      return *this;
}
```

```
#ifndef _HedonicValue_H_
#define _HedonicValue_H_

#define exception class
// public class: HedonicValue
class HedonicValue {
public:
      HedonicValue ();
      HedonicValue (const HedonicValue&);
      virtual ~HedonicValue ();
      HedonicValue& operator = (const HedonicValue &arg);

// Attribute Get/Set Methods:

// Attributes:
private:
      integer     RefHedonicCatagoryID;
      double      HedonicScore;
};
#endif
```

```
public class HedonicValue {
      public HedonicValue () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:

// Attributes:
      private integer RefHedonicCatagoryID;
      private double HedonicScore;
}
```

```
// Implementation of: public class HedonicVector

#include "HedonicVector.h"

// ---------------------------------------------------------
HedonicVector::HedonicVector ()
{
}

// ---------------------------------------------------------
HedonicVector::HedonicVector (const HedonicVector&)
{
}

// ---------------------------------------------------------
HedonicVector::~HedonicVector ()
{
}

// ---------------------------------------------------------
HedonicVector& HedonicVector::operator = (const HedonicVector &arg)
{
      return *this;
}
```

```
#ifndef _HedonicVector_H_
#define _HedonicVector_H_

#define exception class
#include "HedonicValue.h"

// public class: HedonicVector
class HedonicVector {
public:
      HedonicVector ();
      HedonicVector (const HedonicVector&);
      virtual ~HedonicVector ();
      HedonicVector& operator = (const HedonicVector &arg);

// Attribute Get/Set Methods:

// Attributes:
private:
      integer     ReferenceSetID;
      boolean     Preferred;

// Relationships:
public:
      HedonicValue      *unnamed;
};
#endif
```

```
public class HedonicVector {
      public HedonicVector () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:

// Attributes:
      private integer ReferenceSetID;
      private boolean Preferred;

// Relationships:
      public HedonicValue unnamed;
}
```

```
// Implementation of: public class PaletteValue

#include "PaletteValue.h"

// ---------------------------------------------------------------
PaletteValue::PaletteValue ()
{
}

// ---------------------------------------------------------------
PaletteValue::PaletteValue (const PaletteValue&)
{
}

// ---------------------------------------------------------------
PaletteValue::~PaletteValue ()
{
}

// ---------------------------------------------------------------
PaletteValue& PaletteValue::operator = (const PaletteValue &arg)
{
      return *this;
}
```

```
#ifndef _PaletteValue_H_
#define _PaletteValue_H_

#define exception class
// public class: PaletteValue
class PaletteValue {
public:
      PaletteValue ();
      PaletteValue (const PaletteValue&);
      virtual ~PaletteValue ();
      PaletteValue& operator = (const PaletteValue &arg);

// Attribute Get/Set Methods:

// Attributes:
private:
      integer     RefWellID;
      double      Quantity;
};
#endif
```

```
public class PaletteValue {
      public PaletteValue () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:

// Attributes:
      private integer RefWellID;
      private double Quantity;
}
```

```
// Implementation of: public class PaletteVector

#include "PaletteVector.h"

// ---------------------------------------------------------------
PaletteVector::PaletteVector ()
{
}

// ---------------------------------------------------------------
PaletteVector::PaletteVector (const PaletteVector&)
{
}

// ---------------------------------------------------------------
PaletteVector::~PaletteVector ()
{
}

// ---------------------------------------------------------------
PaletteVector& PaletteVector::operator = (const PaletteVector &arg)
{
      return *this;
}
```

```
#ifndef _PaletteVector_H_
#define _PaletteVector_H_

#define exception class
#include "PaletteValue.h"

// public class: PaletteVector
class PaletteVector {
public:
      PaletteVector ();
      PaletteVector (const PaletteVector&);
      virtual ~PaletteVector ();
      PaletteVector& operator = (const PaletteVector &arg);

// Attribute Get/Set Methods:

// Attributes:
private:
      integer     ReferenceSetID;
      boolean     Preferred;

// Relationships:
public:
      PaletteValue      *unnamed;
};
#endif
```

```
public class PaletteVector {
      public PaletteVector () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:

// Attributes:
      private integer ReferenceSetID;
      private boolean Preferred;

// Relationships:
      public PaletteValue unnamed;
}
```

```
// Implementation of: public class PolymerReactivityValue

#include "PolymerReactivityValue.h"

// ---------------------------------------------------------
PolymerReactivityValue::PolymerReactivityValue ()
{
}

// ---------------------------------------------------------
PolymerReactivityValue::PolymerReactivityValue (const
PolymerReactivityValue&)
{
}

// ---------------------------------------------------------
PolymerReactivityValue::~PolymerReactivityValue ()
{
}

// ---------------------------------------------------------
PolymerReactivityValue& PolymerReactivityValue::operator = (const
PolymerReactivityValue &arg)
{
      return *this;
}
```

```
#ifndef _PolymerReactivityValue_H_
#define _PolymerReactivityValue_H_

#define exception class
// public class: PolymerReactivityValue
class PolymerReactivityValue {
public:
      PolymerReactivityValue ();
      PolymerReactivityValue (const PolymerReactivityValue&);
      virtual ~PolymerReactivityValue ();
      PolymerReactivityValue& operator = (const PolymerReactivityValue
&arg);

// Attribute Get/Set Methods:
      const integer&    GetRefPolymerID () const      {return
RefPolymerID;}
      void  SetRefPolymerID (const integer &val)      {RefPolymerID = val;}

// Attributes:
private:
      integer     RefPolymerID;
      double      ReactivityScore;
};
#endif
```

```
public class PolymerReactivityValue {
      public PolymerReactivityValue () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:
      public integer GetRefPolymerID ()    {return RefPolymerID;}
      public void SetRefPolymerID (integer val) {RefPolymerID = val;}

// Attributes:
      private integer RefPolymerID;
      private double ReactivityScore;
}
```

```
// Implementation of: public class PolymerReactivityVector

#include "PolymerReactivityVector.h"

// ---------------------------------------------------------
PolymerReactivityVector::PolymerReactivityVector ()
{
}

// ---------------------------------------------------------
PolymerReactivityVector::PolymerReactivityVector (const
PolymerReactivityVector&)
{
}

// ---------------------------------------------------------
PolymerReactivityVector::~PolymerReactivityVector ()
{
}

// ---------------------------------------------------------
PolymerReactivityVector& PolymerReactivityVector::operator = (const
PolymerReactivityVector &arg)
{
      return *this;
}
```

```
#ifndef _PolymerReactivityVector_H_
#define _PolymerReactivityVector_H_

#define exception class
#include "PolymerReactivityValue.h"

// public class: PolymerReactivityVector
class PolymerReactivityVector {
public:
      PolymerReactivityVector ();
      PolymerReactivityVector (const PolymerReactivityVector&);
      virtual ~PolymerReactivityVector ();
      PolymerReactivityVector& operator = (const PolymerReactivityVector
&arg);

// Attribute Get/Set Methods:

// Attributes:
private:
            ReferenceSetID;
            Preferred;

// Relationships:
public:
      PolymerReactivityValue  *unnamed;
};
#endif
```

```
public class PolymerReactivityVector {
      public PolymerReactivityVector () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:

// Attributes:
      private  ReferenceSetID;
      private  Preferred;

// Relationships:
      public PolymerReactivityValue unnamed;
}
```

```
// Implementation of: public class ReceptorScreeningValue

#include "ReceptorScreeningValue.h"

// ------------------------------------------------------------
ReceptorScreeningValue::ReceptorScreeningValue ()
{
}

// ------------------------------------------------------------
ReceptorScreeningValue::ReceptorScreeningValue (const
ReceptorScreeningValue&)
{
}

// ------------------------------------------------------------
ReceptorScreeningValue::~ReceptorScreeningValue ()
{
}

// ------------------------------------------------------------
ReceptorScreeningValue& ReceptorScreeningValue::operator = (const
ReceptorScreeningValue &arg)
{
      return *this;
}
```

```
#ifndef _ReceptorScreeningValue_H_
#define _ReceptorScreeningValue_H_

#define exception class
// public class: ReceptorScreeningValue
class ReceptorScreeningValue {
public:
      ReceptorScreeningValue ();
      ReceptorScreeningValue (const ReceptorScreeningValue&);
      virtual ~ReceptorScreeningValue ();
      ReceptorScreeningValue& operator = (const ReceptorScreeningValue
&arg);

// Attribute Get/Set Methods:
      const integer&    GetRefReceptorID () const     {return
RefReceptorID;}
      void  SetRefReceptorID (const integer &val)     {RefReceptorID =
val;}

// Attributes:
private:
      integer     RefReceptorID;
      double      ScreeningScore;
};
#endif
```

```
public class ReceptorScreeningValue {
      public ReceptorScreeningValue () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:
      public integer GetRefReceptorID ()  {return RefReceptorID;}
      public void SetRefReceptorID (integer val)      {RefReceptorID =
val;}

// Attributes:
      private integer RefReceptorID;
      private double ScreeningScore;
}
```

```
// Implementation of: public class ReceptorScreeningVector

#include "ReceptorScreeningVector.h"

// ---------------------------------------------------------
ReceptorScreeningVector::ReceptorScreeningVector ()
{
}

// ---------------------------------------------------------
ReceptorScreeningVector::ReceptorScreeningVector (const
ReceptorScreeningVector&)
{
}

// ---------------------------------------------------------
ReceptorScreeningVector::~ReceptorScreeningVector ()
{
}

// ---------------------------------------------------------
ReceptorScreeningVector& ReceptorScreeningVector::operator = (const
ReceptorScreeningVector &arg)
{
      return *this;
}
```

```
#ifndef _ReceptorScreeningVector_H_
#define _ReceptorScreeningVector_H_

#define exception class
#include "ReceptorScreeningValue.h"

// public class: ReceptorScreeningVector
class ReceptorScreeningVector {
public:
      ReceptorScreeningVector ();
      ReceptorScreeningVector (const ReceptorScreeningVector&);
      virtual ~ReceptorScreeningVector ();
      ReceptorScreeningVector& operator = (const ReceptorScreeningVector
&arg);

// Attribute Get/Set Methods:

// Attributes:
private:
      integer     ReferenceSetID;
      boolean     Preferred;

// Relationships:
public:
      ReceptorScreeningValue  *unnamed;
};
#endif
```

```
public class ReceptorScreeningVector {
      public ReceptorScreeningVector () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:

// Attributes:
      private integer ReferenceSetID;
      private boolean Preferred;

// Relationships:
      public ReceptorScreeningValue unnamed;
}
```

```
// Implementation of: public class RefDockingValue

#include "RefDockingValue.h"

// ---------------------------------------------------------
RefDockingValue::RefDockingValue ()
{
}

// ---------------------------------------------------------
RefDockingValue::RefDockingValue (const RefDockingValue&)
{
}

// ---------------------------------------------------------
RefDockingValue::~RefDockingValue ()
{
}

// ---------------------------------------------------------
RefDockingValue& RefDockingValue::operator = (const RefDockingValue &arg)
{
      return *this;
}
```

```
#ifndef _RefDockingValue_H_
#define _RefDockingValue_H_

#define exception class
// public class: RefDockingValue
class RefDockingValue {
public:
      RefDockingValue ();
      RefDockingValue (const RefDockingValue&);
      virtual ~RefDockingValue ();
      RefDockingValue& operator = (const RefDockingValue &arg);

// Attribute Get/Set Methods:
      const integer&    GetRefReceptorID () const     {return
RefReceptorID;}
      void  SetRefReceptorID (const integer &val)     {RefReceptorID =
val;}

// Attributes:
private:
      integer     RefReceptorID;
      double[]    DockingScore;
};
#endif
```

```
public class RefDockingValue {
      public RefDockingValue () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:
      public integer GetRefReceptorID ()  {return RefReceptorID;}
      public void SetRefReceptorID (integer val)      {RefReceptorID =
val;}

// Attributes:
      private integer RefReceptorID;
      private double[] DockingScore;
}
```

```
// Implementation of: public class ReceptorDockingVector

#include "ReceptorDockingVector.h"

// ---------------------------------------------------------
ReceptorDockingVector::ReceptorDockingVector ()
{
}

// ---------------------------------------------------------
ReceptorDockingVector::ReceptorDockingVector (const ReceptorDockingVector&)
{
}

// ---------------------------------------------------------
ReceptorDockingVector::~ReceptorDockingVector ()
{
}

// ---------------------------------------------------------
ReceptorDockingVector& ReceptorDockingVector::operator = (const
ReceptorDockingVector &arg)
{
      return *this;
}
```

```
#ifndef _ReceptorDockingVector_H_
#define _ReceptorDockingVector_H_

#define exception class
#include "RefDockingValue.h"

// public class: ReceptorDockingVector
class ReceptorDockingVector {
public:
      ReceptorDockingVector ();
      ReceptorDockingVector (const ReceptorDockingVector&);
      virtual ~ReceptorDockingVector ();
      ReceptorDockingVector& operator = (const ReceptorDockingVector &arg);

// Attribute Get/Set Methods:

// Attributes:
private:
      integer     ReferenceSetID;
      boolean     Preferred;

// Relationships:
public:
      RefDockingValue    *unnamed;
};
#endif
```

```
public class ReceptorDockingVector {
      public ReceptorDockingVector () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:

// Attributes:
      private integer ReferenceSetID;
      private boolean Preferred;

// Relationships:
      public RefDockingValue unnamed;
}
```

```
// Implementation of: public class StructuralVector

#include "StructuralVector.h"

// ----------------------------------------------------------
StructuralVector::StructuralVector ()
{
}

// ----------------------------------------------------------
StructuralVector::StructuralVector (const StructuralVector&)
{
}

// ----------------------------------------------------------
StructuralVector::~StructuralVector ()
{
}

// ----------------------------------------------------------
StructuralVector& StructuralVector::operator = (const StructuralVector
&arg)
{
      return *this;
}
```

```
#ifndef _StructuralVector_H_
#define _StructuralVector_H_

#define exception class
#include "ChemSimilartyValue.h"

// public class: StructuralVector
class StructuralVector {
public:
      StructuralVector ();
      StructuralVector (const StructuralVector&);
      virtual ~StructuralVector ();
      StructuralVector& operator = (const StructuralVector &arg);

// Attribute Get/Set Methods:

// Attributes:
private:
      integer     ReferenceSetID;
      boolean     Preferred;

// Relationships:
public:
      ChemSimilartyValue      *unnamed;
};
#endif
```

```
public class StructuralVector {
      public StructuralVector () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:

// Attributes:
      private integer ReferenceSetID;
      private boolean Preferred;

// Relationships:
      public ChemSimilartyValue unnamed;
}
```

```
// Implementation of: public class Scent

#include "Scent.h"

// ------------------------------------------------------------
Scent::Scent ()
{
}

// ------------------------------------------------------------
Scent::Scent (const Scent&)
{
}

// ------------------------------------------------------------
Scent::~Scent ()
{
}

// ------------------------------------------------------------
Scent& Scent::operator = (const Scent &arg)
{
      return *this;
}
```

```
#ifndef _Scent_H_
#define _Scent_H_

#define exception class
#include "Compound.h"
#include "PaletteVector.h"
#include "ReceptorScreeningVector.h"
#include "PolymerReactivityVector.h"
#include "HedonicVector.h"
#include "Compound.h"

// public class: Scent
class Scent {
public:
      Scent ();
      Scent (const Scent&);
      virtual ~Scent ();
      Scent& operator = (const Scent &arg);

// Attribute Get/Set Methods:
      const String&     GetName () const   {return Name;}
      void  SetName (const String &val)    {Name = val;}

// Attributes:
private:
      String      Name;
      integer     ReminiscentsID;
      Vector      AltenateNames;
      String      Description;
            SensoryIndex;

// Relationships:
public:
      PaletteVector     *unnamed;
      ReceptorScreeningVector *unnamed;
      PolymerReactivityVector *unnamed;
      HedonicVector     *unnamed;
};
#endif
```

```
public class Scent {
      public Scent () {
      }
      protected void finalize () {
      }

// Attribute Get/Set Methods:
      public String GetName ()      {return Name;}
      public void SetName (String val)    {Name = val;}

// Attributes:
      private String Name;
      private integer ReminiscentsID;
      private Vector AltenateNames;
      private String Description;
      private  SensoryIndex;

// Relationships:
      public PaletteVector unnamed;
      public ReceptorScreeningVector unnamed;
      public PolymerReactivityVector unnamed;
      public HedonicVector unnamed;
}
```

APPENDIX B

ScentWare™ SDK

Version 1.00

CONFIDENTIAL

To apply for the ScentWare developers program,
go to the DigiScents web site and follow the developers link.

http://www.digiscents.com

CONFIDENTIAL

Table of Contents

Quickstart 3
Security Key 4
Theory of Operations 5
Code samples and boilerplate 7
Virtual ISmell emulator 8
Library Reference 9
close 10
emitScent 11
emitScent 12
errorExists 13
getDriverVersion 14
getErrors 15
getVersion 16
isDevicePresent 17
isOpen 18
isPersistent 19
isPersistent 20
open 21
setCallback 22
setComPort 23
setDevicePresent 24
setDiffuser 25
setScentStatus 26
setVaporizer 27
Appendix 28
Figure 1.0 Types 29
Figure 2.0 ssBool 30
Figure 3.1 ssSdkError 31
Figure 3.2 ssThreadError 32
Figure 3.3 ssDriverError 33
Figure 3.4 ssCommIoError 34
Figure 3.5 ssUsbIoError 35
Figure 3.6 ssErrorStruct 36
Figure 4.0 ssCallback 37
Figure 5.0 ssIntensity 38
Figure 6.0 ssScent 39
Figure 7.0 iScentWareSDK 40
Figure 7.0 Boilerplate 41

Copyright (c) 2000, DigiScents, Inc.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means, without the prior written permission of the publisher. The author and publisher have made every effort in the preparation of this book to ensure the accuracy of the information. However, the information contained in this book is sold without warranty, either express or implied. Neither the author nor DigiScents will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

DigiScents™, iSmell™, Snarfal™, ScentStream™, ScentWare™, ReminiScents™, ScentPalette™, ScentMixer™, ScentTracker™, ScentMail™, ScentTracks™, ScentObjects™, ScentSignatures™, ScentSpace™, and ScentBrowser™ are all trademarks of DigiScents, Inc. All other companies or products listed herein are trademarks or registered trademarks of their respective owners.

CONFIDENTIAL

Quick Start for the ScentWare SDK

1) Run the SDK installer. The installer will install two DLL's into the windows\system or winnt\ directory. The DLL's are ScentWareSDK.dll and iSmellDriver.dll. They will be registered by the installer.

2) Add CreateScentWareSDK.cpp to your project. Add the path to the ScentWareSDK header to your projects include paths. The path should be \ScentWareSDK\include if you did not change the installation directory.

3) Add global variables and header file ScentWareSDK.h.

```
// global variables
IScentWareSDK * theScentWareSDK = NULL;
ssScent cinnamon;
```

4) Add the following code to your startup routine.

A) If **NOT** using CoInitialize() and CoUninitialize()

```
// create SDK
if((theScentWareSDK=CreateScentWare())==NULL)
      {
      // the ScentWareSDK DLL is not properly installed, fail.
      }
```

B) If Using CoInitialize/CoUninitialize()

// CoInitialize() must be called prior to calling CoCreateScentWare()

```
// create SDK
if((theScentWareSDK=CoCreateScentWare())==NULL)
      {
      // the ScentWareSDK DLL is not properly installed.
      // fail
      }
```

5) Add the following code to your shutdown routine prior to the call to CoUninitialize() if it is being used.

```
DestroyScentWare(theScentWareSDK);
```

CONFIDENTIAL

6) Load a scent object from disk. The scent objects will be found in the ScentWareSDK\scents directory.

```
YourLoad(&cinnamon,sizeof(ssScent));        // use your file system or asset picker
```

7) Make the following call in an appropriate place.

```
TheScentWareSDK->emitScent(cinnamon);
```

8) If an iSmell or the emulator is connected to your computer, a scent will be emitted from the iSmell device or message will be given by the emulator.

ScentWare Security

Because ScentWare is being proliferated via the Internet, the data objects, ScentObjects™, have a built in security key. The SDK will not emit scents unless the ScentObject contain a valid key, emitScent will fail and issue an ssInvalidKey error message.

Scents that do not contain keys are called unauthorized scents. Scents with embedded keys are authorized scents. Unauthorized scents can be emitted using the *ReminiScents™ ScentBrowser™*.

Developers will be issued a unique key with their license agreement. This key will be used to authorize the scent objects that are to be redistributed in product. It is the developer's responsibility to aggregate the individual scent objects for distribution in product.

The SDK will reject scents that do not have vaild keys so it is important for the developer to ensure that they authorize the scent that they wish to use in the product.

Note: The 1.0 SDK is distributed with a set of fully authorized dummy ScentObjects. These objects are for testing with the emulator only and should be replaced for use with an actual iSmell device.

CONFIDENTIAL

Theory of Operations

The ScentWare SDK is a COM model object packaged as a DLL for use as an in process server. It may be handled as a standard Windows COM object, and it exports the requisite functions.

The SDK has a simple flow of execution. At start run the object is created using either the boilerplate helper function IScentWareSDK *CreateScentWare(void) or CoCreateScentWare() if you are using CoInitialize() and CoUninitialize().

At end run the SDK object is destroyed once again with a boilerplate helper function ssBool DestroyScentWareSDK(IScentWareSDK * theScentWareSDK). This function simply calls the SDK's Release() method.

Once the object has been created the environment is opened and closed using the open() and close() member functions.

After a successful call to open() has been made, it is possible to call one of the emitScent(...) function signatures or the accessor and mutator functions of the ScentWare object.

When the application is finished using the ScentWare SDK, a call is made to close().

A function is supplied to allow the app to set the iSmell's communications port prior to the call to open(), in this case, setComPort(). This call will change, as the production iSmell device will use USB.

The version accessor functions will allow the app to obtain version info on both the SDK as well as the iSmell driver that is found on the runtime machine. If no driver is found then the getDriverVersion() function will return NULL.

The SDK runs on top of a thread that does the work of emitting a scent and timing its persistence envelope. This thread has a lower then normal priority. The rendering thread also prevents the emission of a given scent if the same scent is still persistent.

If an emit call is made for a different scent while the iSmell device is in busy mode i.e. the device actually emitting scent essence, the scent will be held pending until the device is not busy and the scent will be emitted. It is only possible to hold a single scent pending at any given time.

The SDK does not call CoInitialize(),CoInitializeEx() or CoUninitialize(); this is the responsibility of your application, if you are using this family of functions. Note that these functions are called in the main() procedure of the example if enabled. All memory use by the SDK is automatic data. No memory is allocated by the SDK or driver.

CONFIDENTIAL

The iSmell™ device operates by emitting small portions of scent essences. The vaporizer vaporizes the essence mixture, which is then diffused into the surrounding air by the diffuser.

Each scent has vaporizer and diffuser settings in the object, so that it is not necessary to specifically call setVaporizer() or setDiffuser(). An overloaded signature of emitScent() is available. This allows the calling application full control of these settings.

The intensity of a scent is based on several factors, the foremost of which is the amount of essence that is emitted. Vaporizer and Diffuser settings can also affect the intensity but to a lesser degree than the amount of essence emitted. The SDK settings for intensity deal with the amount of essence emitted by the iSmell.

A scent can be either simple or complex; a simple scent uses only a single scent essence while a complex scent uses up to eight different essences. Complex scents are a product of several different essences emitted with varying intensities. Both types of scents have singular Vaporization and Diffusion settings for their intensity level, ssMild, ssNormal and ssStrong.

A scent has two phases of existence: emission, and persistence. While the scent is being emitted by the iSmell device, the device is busy and cannot accept further commands. If a command for a dissimilar scent to the current scent is received during emission time it will be held pending. While the scent is persistent in the air, the thread mechanism disallows the iSmell device from producing further emissions of the same scent.

This prevents a succession of game space object triggers from emitting an overwhelming scent when only a single emission is needed. Iteratively calling emitScent() without the protection of the threads persistence aging process can overload the vaporizer and produce a strong scent with an overly long persistence envelope.

The command slot for the thread is a single command deep. If a command is already pending, successive calls will fail. When the thread accepts the command it clears the command slot and makes it ready for the next command.

In scent enabling an interactive application, generally, "less is more". A world scent that is triggered every 5-10 minutes coupled with prize, event or entity based scents is one potential design pattern.

Because of the nature of scent persistence the calling application should be discreet about the number of calls made. The thread based protection from emitting multiple iterations of the same scent will offer further protection from excessive triggering of scents.

CONFIDENTIAL

Code Samples and Boilerplate

Code samples and boilerplate are supplied with the SDK that contain some C type functions that will quick start and demonstrate the use of the SDK.

```
IScentWareSDK *CreateScentWare(void);
```

Creates the COM object SDK without calling CoCreateInstance(). When using this Creation routine it is unnecessary to call CoInitialize() or CoUnInitialize().

```
IScentWareSDK *CoCreateScentWare(void);
```

Creates the COM object SDK using the win32 call CoCreateInstance(). It is necessary for the application to call CoInitialize() prior to using this routine and CoUninitialize() at end run.

```
ssBool DestroyScentWare(IScentWareSDK *theScentWareSDK);
```

Destroys the COM object using the Release() method of the SDK. If the SDK object is NULL, this routine will fail.

```
ssVoid ShowScentWareErrors(IScentWareSDK *theScentWareSDK);
```

Will show errors generated by the SDK as text messages. If no error has occurred the routine will return. This routine uses cout to display the errors.

```
ssStringPtr GetScentWareErrorString(IScentWareSDK *theScentWareSDK);
```

Will return a pointer to a static buffer containing the current error string. If there is no error the function will return NULL.

```
ssBool IsPlatformScentEnabled(ssVoid);
```

Will return ssTrue if the runtime platform is scent enabled or ssFalse if it is not.

A simple application "smelloWorld.cpp" is supplied that demonstrates the initialization and usage of the SDK.

CONFIDENTIAL

Software Emulator - Virtual iSmell device

A Windows software emulator is supplied that, when connected to the host station via a null modem cable, will log and display all transactions with the virtual iSmell. This application and its support files are located in ScentWareSDK\Emulator

The emulator requires a second computer (the target) to be connected to the host (development) station via the null modem cable. This cable should be connected from COM1 on the host to COM1 on the target.

Once the connection is made, run the program iSmellEmulator.exe on the target machine and your scent enabled application on the host. The emulator will output the commands sent to it as text in a text window.

The emulator program uses a list of scent names supplied in names.txt, this file can be customized to meet your specific needs. The file format is a simple text file containing the names of each scent on a separate line with the word End followed by a return at the end of the listing.

CONFIDENTIAL

ScentWare Library Reference

Prototype: ssBool ScentWareSDK::close(ssVoid)

Arguments: void

Returns: ssTrue, on success
or
the runtime platform is not equipped with an iSmell device.
ssFalse, on fail
the iSmell driver reported an error.
or
the rendering thread was unable to terminate.

Description:

Closes the iSmell driver, destroys the rendering thread and releases all COM objects.

Notes:

If the driver is not present this will return ssTrue.

Example:

```
if(theScentWareSDK->close()==ssFalse)      // close the SDK
      {
      OutputDebugString("*ERROR* theScentWareSDK->close() FALSE\n");

      ShowScentWareErrors(theScentWareSDK);      // show any errors

      return false;
      }
```

CONFIDENTIAL emitScent

Prototype: ssBool ScentWareSDK::emitScent(ssScentPtr scentObject,
ssIntensity intensity=ssNormal)

Arguments: ssScentPtr scentObject, a pointer to a scent object.
ssIntensity intensity, ssMild, ssNormal, ssStrong, this defaults to ssNormal.
this relates to the intensity of the scent emitted.

Returns: ssTrue, on success
or
the runtime platform is not equipped with an iSmell device.
or
the scent has been recently emitted and is already persistant in the air.

ssFalse, on fail
an emitScent() command is in the pipe waiting for render thread service.
or
scentObject is NULL

Description:

One of the function signatures for emitting a scent. The arg intensity defaults to ssNormal intensity and can be omitted for this behavior.

Notes:

If intensity is out of range an SDK error will be posted, the parameter will be set to ssNormal and execution will continue. See Also: emitScent(ssScentPtr scentObject,ssIntensity intensity,ssU16 diffusion,ssU16 vaporization)

Example:

```
if(theScentWareSDK->emitScent(&cinnamon)==ssFalse)          // emit normal scent
      {
      OutputDebugString("ssEmitScent(&cinnamon) FAIL");
      ShowScentWareErrors(theScentWareSDK);                 // show any errors
      return false;
      }


if(theScentWareSDK->emitScent(&cinnamon,ssMild)==ssFalse)   // emit mild scent
      {
      OutputDebugString("ssEmitScent(&cinnamon) FAIL");
      ShowScentWareErrors(theScentWareSDK);                       // show any
errors
      return false;
      }
```

CONFIDENTIAL emitScent

Prototype: ssBool ScentWareSDK::emitScent(ssScentPtr scentObject,
ssIntensity intensity,
ssU16 diffusion,
ssU16 vaporization)

Arguments: ssScentPtr scentObject, a pointer to a scent object.
ssIntensity intensity, ssMild, ssNormal or ssStrong
ssU16 diffusion; (0-255) controls the rate of diffusion into the air.
ssU16 vaporization (0-kMaxVaporizerSetting) controls the rate of vaporization into the air.

Returns: ssTrue, on success
or
the runtime platform is not equipped with an iSmell device.
or
the scent has been recently emitted and is already persistant in the air.

ssFalse, on fail
an emitScent() command is in the pipe waiting for render thread service.
or
scentObject is NULL

Description:

One of the function signatures for emitting a scent. The arguments allow a greater degree of control over how the scent is emitted by the iSmell. Intensity controls the quantity of scent essence emitted, diffusion controls how quickly the iSmell pushes the vaporized essence into the surrounding environment and vaporization controls how quickly the scent essence is vaporized.

Notes:

If intensity or vaporizer are out of range an SDK error will be posted, the parameter will be set to either ssNormal for intensity or kDefaultVaporizerSetting for vaporizer and execution will continue. See Also: emitScent(ssScentPtr scentObject,ssIntensity intensity)

Example:

```
if(theScentWareSDK->emitScent(&cinnamon,ssStrong,255,100)==ssFalse)
      {
      cout << "ssEmitScent(&cinnamon,ssStrong,255,100) FAIL" << endl;
      ShowScentWareErrors(theScentWareSDK);      // show any errors
      return false;
      }
```

CONFIDENTIAL errorExists

Prototype: ssBool ScentWareSDK::errorExists(ssVoid)

Arguments: none

Returns: ssTrue if an error has occurred.
ssFalse if no error exists.

Description:

Used to determine if an error has occured in the ScentWareSDK, there is a helper routine in the boilerplate called ShowScentWareErrors() that will show the error numbers as text messages.

Notes:

None

Example:

```
if(theScentWareSDK->errorExists()==ssTrue)      // has a error occured ?
      {
      ShowScentWareErrors(theScentWareSDK);     // show any errors (boilerplate)
      }
```

CONFIDENTIAL getDriverVersion

Prototype: const char *ScentWareSDK::getDriverVersion(ssVoid)

Arguments: void

Returns: a pointer to a NULL terminated string containing the iSmell driver version.
or
NULL if the iSmell driver is not present on the runtime machine.

Description:

Used to obtain the drivers version string.

Notes:

This must be called post to the call to open().

Example:

```
const char *driverVersion=theScentWareSDK->getDriverVersion();
OutputDebugString(driverVersion);
```

CONFIDENTIAL getErrors

Prototype: ssErrorStructPtr ScentWareSDK::getErrors(ssVoid)

Arguments: void

Returns: a pointer to the SDK's error structure.

Description:

Allows the app to get a pointer to the structure in which the SDK stores its errors.

Notes:

This only needs to be called once to allow the application developer to view the error structure which is a private member of the class.

Example:

```
// get the pointer to the errors structure
ssErrorStructPtr myScentWareErrors = theScentWareSDK->getErrors();
```

Prototype: const char * ScentWareSDK::getVersion(ssVoid)

Arguments: void

Returns: a pointer to a const string containing the SDK version number.

Description:

Used to obtain a string containing the version info for the SDK.

Notes:

None

Example:

```
const char *theVersion = theScentWareSDK->getVersion();      // get SDK version
```

Prototype: ssBool ScentWareSDK::isDevicePresent(ssVoid)

Arguments: void

Returns: ssTrue if an iSmell or emulator is connected to the system.
ssFalse if an iSmell or emulator is not connected.

Description:

Used to determine if an iSmell is present on the runtime system. This will allow the app to enable or disable any preference controls that relate to scent.

Notes:

Must be called post the call to open().

Example:

```
if(theScentWareSDK->isDevicePresent()==ssFalse)
      {
      // disable controls
      }
else
      {
      // enable controls
      }
```

Prototype: ssBool ScentWareSDK::isOpen(ssVoid)

Arguments: void

Returns: ssTrue if the SDK is open.
ssFalse if the SDK is not open.

Description:

Allows the app to determine if the iSmellDriver has been opened.

Notes:

If the driver is not present this will return ssTrue.

Example:

```
if(theScentWareSDK->isOpen()==ssTrue)      // is the iSmellDrive open ?
      {
      if(theScentWareSDK->isDevicePresent()==ssFalse)
            {
            // disable controls
            }
      else
            {
            // enable controls
            }
      )
```

CONFIDENTIAL isPersistent

Prototype: ssBool ScentWareSDK::isPersistent(ssScentPtr scentObject)

Arguments: ssScentPtr scentObject, a pointer to a scent object.

Returns: ssTrue if the given scent is currently persistent.
ssFalse if the given scent is not currently persistent.
or
scentObject is NULL

Description:

Used to determine if the given scent is currently persistent.

Notes:

Persistence is the emission time plus the scents local persistence time.

Example:

```
if(theScentWareSDK->isPersistent(worldScent)==ssFalse)
      {
      theScentWareSDK->emitScent(worldScent);   // emit a scent
      }
```

Prototype: ssBool ScentWareSDK::isPersistent(ssVoid)

Arguments: void

Returns: ssTrue if any scent is currently persistent.
ssFalse if no scent is not currently persistent.

Description:

Used to determine if any scent is currently persistent.

Notes:

Persistence is the emission time plus the scents local persistence time.

Example:

```
if(theScentWareSDK->isPersistent()==ssFalse)
      {
      theScentWareSDK->emitScent(cinnamon);      // emit a scent
      }
```

Prototype: ssBool ScentWareSDK::open(ssVoid)

Arguments: void

Returns: ssTrue, on success
or
the runtime platform is not equipped with an iSmell.
ssFalse, an error occured opening the iSmell driver
or
starting the rendering thread.

Description:

Opens the ScentWareSDK and the iSmell driver if one is present.

Notes:

The ScentWareSDK will function normally and show no errors if the runtime platform is not equipped with an iSmell device.

Example:

```
if(theScentWareSDK->open()==ssFalse)        // open the SDK
      {
      OutputDebugString("*ERROR* theScentWareSDK->open() FALSE\n");

      ShowScentWareErrors(theScentWareSDK);      // show any errors

      return false;
      }
```

CONFIDENTIAL **setCallback**

Prototype: ssBool ScentWareSDK::setCallback(ssScentPtr scentObject,
ssCallback callback,
ssS32 userId)

Arguments: ssScentPtr scentObject, a pointer to a scent object.
ssCallback callback, the address of the callback function.
ssS32 usrId, a number that the app will use to identify the caller.

Returns: ssTrue if the driver is not present.
or
if the scent object was set with the callback and ID.

ssFalse if scentObject is NULL
or
callback is NULL
or
scentObject is currently persistent

Description:

Initializes a scent object with a callback function and ID. Upon completion of the scents persistence period the callback function will be invoked.

Notes:

The callback and userId members of the scent will not be cleared during the callback process. This is to create a situation where the SDK does not retain pointers to scent objects for the purpose of writing back to them, the retained pointers are used for observation only.

Example:

```
void MyCallback(ssS32 ID)      { // do something here }      // declare callback

// set callback
if(theScentWareSDK->setCallback(&cinnamon,MyCallback,kMyCallbackID)==ssFalse)
      {
      OutputDebugString("setCallback(&cinnamon,MyCallback,kMyCallbackID) FAIL");
      ShowScentWareErrors(theScentWareSDK);                - // show any errors
      return false;
      }

// emit scent using any of the emitScent function signatures
if(theScentWareSDK->emitScent(&cinnamon)==ssFalse)
      {
      OutputDebugString("ssEmitScent(&cinnamon) FAIL");
      ShowScentWareErrors(theScentWareSDK);                // show any errors
      return false;
      }
```

CONFIDENTIAL setComPort

Prototype: ssBool ScentWareSDK::setComPort(ssStringPtr portName)

Arguments: portName, a NULL terminated string containing the port name.

Returns: ssTrue, on success
ssFalse, if the SDK is already open.
or
if the string is incorrectly formatted.
or
portName is NULL

Description:

Used for the comm port type devices (see note) to set which comm port will be used to communicate with the iSmell. Valid portNames are "COM1:", "COM2:" "COM3:" etc.

Notes:

This must be called prior to the call to open(). The production iSmell's will be USB rather then serial devices, as a result of this differential this API is subject to change to allow the selection of a USB device.

Example:

```
if(theScentWareSDK->setComPort("COM1:")==ssFalse)      // set the com port
      {
      OutputDebugString("*ERROR* theScentWareSDK->setComPort(\"COM1:\")
FALSE\n");

      ShowScentWareErrors(theScentWareSDK);                  // show any errors

      return false;
      }
```

CONFIDENTIAL setDevicePresent

Prototype: ssBool ScentWareSDK::setDevicePresent(ssBool yesOrNo)

Arguments: yesOrNo, ssTrue

Returns: ssTrue if the driver is not present.
or
if the flag was set.

ssFalse if the driver is already open.
or
yesOrNo is out of range (ssTrue or ssFalse)

Description:

Allows the system to run on a platform that has the driver installed but does not have either an iSmell or the virtual iSmell emulator attached.

Notes:

None

Example:

```
if(theScentWareSDK->setDevicePresent(ssFalse)==ssTrue)      // disable the device
      (
      OutputDebugString("iSmell disabled");
      )
```

CONFIDENTIAL setDiffuser

Prototype: ssBool ScentWareSDK::setDiffuser(ssU16 diffusionRate)

Arguments: ssU16 diffusionRate, (0-255) the rate of the diffuser in the iSmell.

Returns: ssTrue, on success
or
the runtime platform is not equipped with an iSmell device.
ssFalse, on fail
the driver reported an error. (device not on or connected)

Description:

Allows setting of the diffuser in the iSmell independant of the emission of a scent.

Notes:

If diffusionRate is out of range (0-255) it will be set to kDefaultDiffuserSetting and execution will continue.

Example:

```
if(theScentWareSDK->setDiffuser(245)==ssFalse)  // set the diffuser
      {
      OutputDebugString("theScentWareSDK->setDiffuser(245) FALSE\n");

      ShowScentWareErrors(theScentWareSDK);     // show any errors

      return false;
      }
```

CONFIDENTIAL setScentStatus

Prototype: ssBool ScentWareSDK::setScentStatus(ssBool onOrOff)

Arguments: ssBool onOrOff, ssTrue or ssFalse

Returns: ssTrue on success
ssFalse on fail, system not open.
or
onOrOff is out of range (ssTrue-ssFalse)

Description:

Used to enable and disable the emission of scents from the iSmell.

Notes:

If the driver is not present this will return ssTrue.

Example:

```
if(scentOnOrOffBox->GetCheck()==true)
      theScentWareSDK->setScentStatus(ssTrue);  // turn scent on
else
      theScentWareSDK->setScentStatus(ssFalse); // turn scent off
```

Prototype: ssBool ScentWareSDK::setVaporizer(ssU16 vaporizationRate)

Arguments: ssU16 vaporizationRate, (0-kMaxVaporizerSetting) the vaporizer temperature.

Returns: ssTrue, on success
or
the runtime platform is not equipped with an iSmell device.
ssFalse, on fail
the driver reported an error. (device not on or connected)

Description:

Allows setting of the vaporizer in the iSmell independent of the emission of a scent.

Notes:

if vaporizationRate is greater then kMaxVaporizerSetting it will be set to the value kDefaultVaporizerSetting and execution will continue.

Example:

```
if(theScentWareSDK->setVaporizer(75)==ssFalse)  // set the vaporizer
      (
      OutputDebugString("theScentWareSDK->setVaporizer(75) FALSE\n");

      ShowScentWareErrors(theScentWareSDK);     // show errors, boilerplat

      return false;
      }
```

CONFIDENTIAL

Appendix

CONFIDENTIAL

Figure 1.0 Types

```
typedef     void                    ssVoid;
typedef     *                       ssPtr;
typedef     **                      ssHdl;
typedef     signed char             ssS8;
typedef     ssS8  *                 ssS8Ptr;
typedef     unsigned char           ssU8;
typedef     ssU8 *                  ssU8Ptr;
typedef     signed short int        ssS16;
typedef     ssS16 *                 ssS16Ptr;
typedef     unsigned short int      ssU16;
typedef     ssU16 *                 ssU16Ptr;
typedef     signed long int         ssS32;
typedef     ssS32 *                 ssS32Ptr;
typedef     unsigned long int       ssU32;
typedef     ssU32 *                 ssU32Ptr;
typedef     char                    ssString;
typedef     ssString *              ssStringPtr;
typedef     float                   ssFloat;
typedef     ssFloat *               ssFloatPtr;
typedef     ssU32                   ssSystemId;
```

CONFIDENTIAL

Figure 2.0 ssBool

```
typedef enum
   {
   ssFalse  =     0,    // false
   ssTrue   =     1     // true
   }ssBool;
```

CONFIDENTIAL

Figure 3.1 ssSdkError

```
typedef enum
   {
   ssNoError                          =     0,    // all is well
   ssDeviceNotInstalled               =     1,    // iSmellDriver not installed
   ssOpenFail                         =     2,    // open failed
   ssDriverMustBeClosedToSetComPort   =     3,    // cant set if open
   ssComPortStringIsBad               =     4,    // "COM1:" "COM2:" etc...
   ssUnableToCreateThread             =     5,    // can't create thread
   ssUnsupportedInterface             =     6,    // unsupported interface request
   ssParameterIsOutOfRange            =     7,    // parameter is out of range
   ssParameterIsOutOfRangeFixed       =     8,    // out of range parameter fixed
   ssNullParameter                    =     9,    // NULL parameter error
   ssInvalidKey                       =     10,   // invalid security key
   ssSeeDriverError                   =     11,   // driver error occured
   ssLastError
   }ssSdkError;
```

CONFIDENTIAL

Figure 3.2 ssThreadError

```
typedef enum
   {
   ssThreadNoError                  =     0,    // groovatious
   ssThreadCantGetExitCode          =     1,    // GetExitCodeThread() failed
   ssThreadCantTerminate            =     2,    // TerminateThread() failed
   ssThreadNoAgingEntriesAvailable  =     3,    // > 16 scents persistent
   ssThreadUnableToSetPriority      =     4,    // SetThreadPriority() failed
   ssThreadSeeDriverError           =     5,    // driver error in thread FSM
   ssThreadLastError
   }ssThreadError;
```

CONFIDENTIAL

Figure 3.3 ssDriverError

```
typedef enum
   {
   ssDriverNoError                  =     0,    // all is well
   ssDriverUnsupportedInterface     =     1,    // IID_IUnknown/IID_iSmellDriver
   ssDriverDeviceIsNotOpen          =     2,    // didn't call open()
   ssDriverSpuriousCallToOpen       =     3,    // called open() more then once
   ssDriverSpuriousCallToClose      =     4,    // called close more then once
   ssDriverCellNumberOutOfRange     =     5,    // cell number is out of range
   ssDriverCheckSumError            =     6,    // checksum error
   ssDriverCanOnlyMarkEightCells    =     7,    // only eight cells can be marked
   ssDriverRowNumberOutOfRange      =     8,    // valid rows are 0-7
   ssDriverColNumberOutOfRange      =     9,    // valid cols are 0-15
   ssDriverDidntRxStatusString      =     10,   // didnt get a status string
   ssDriverNotCheckSumMessage       =     11,   // invalid command confirmation
   ssDriverCantOpenCommPort         =     12,   // can't open COMM port
   ssDriverComPortStringTooLong     =     13,   // "COM1:" "COM2:" ...
   ssDriverComPortStringCorrupt     =     14,   // "COM1:" "COM2:" ...
   ssDriverNullArgument             =     15,   // NULL argument passed to driver
   ssDriverLastError
   }ssDriverError;
```

CONFIDENTIAL

Figure 3.4 ssCommIoError

```
typedef enum
   {
   ssCommNoError                          =     0,    // cool.
   ssCommFailedToOpenCommPort             =     1,    // CreateFile() failed
   ssCommFailedToSetCommConfiguration     =     2,    // SetCommConfig() failed
   ssCommFailedToSetCommTimeOuts          =     3,    // SetCommTimeouts() failed
   ssCommFailedToGetCommState             =     4,    // GetCommState() failed
   ssCommFailedToSetCommState             =     5,    // SetCommState() failed
   ssCommFailedToWriteToPort              =     6,    // WriteFile() failed
   ssCommFailedToReadFromPort             =     7,    // ReadFile() failed
   ssCommFailedToClearError               =     8,    // ClearCommError() failed
   ssCommFailedToClearRxBuffer            =     9,    // PurgeComm(RX) failed
   ssCommLastError
   }ssCommIoError;
```

CONFIDENTIAL

Figure 3.5 ssUsbIoError

```
typedef enum
   {
   ssUsbNoError   =     0,
   ssUsbLastError
   }ssUsbIoError;
```

CONFIDENTIAL

Figure 3.6 ssErrorStruct

```
typedef struct
   {
   ssSdkError     sdkError;          // SDK errors
   ssThreadError  threadError;       // Thread errors
   ssDriverError  driverError;       // iSmellDriver.dll errors
   ssCommIoError  commIoError;       // COMM IO errors
   ssUsbIoError   usbIoError;        // USB IO errors
   }ssErrorStruct;

typedef ssErrorStruct * ssErrorStructPtr;
```

CONFIDENTIAL

Figure 4.0 ssCallback

```
typedef void (*ssCallback)(ssS32);
```

CONFIDENTIAL

Figure 5.0 ssIntensity

```
typedef enum
   {
   ssMild              =     0,
   ssNormal            =     1,
   ssStrong            =     2,
   ssLastIntensity     =     3
   }ssIntensity;
```

CONFIDENTIAL

Figure 6.0 ssScent

```
typedef struct
   {
   ssString       name[kMaxScentNameLength];    // the components common name
   ssU16          cell;                         // the cell containing the scent
   ssU16          intensity;                    // the normal intensity
   ssSystemId     systemId;                     // system scentId
   }ssScentComponent;

typedef struct
   {
   ssU16                revision;                        // rev number (reserved)
   ssString             name[kMaxScentNameLength];       // top level common name
   ssU16                diffuser;                        // diffusion rate
   ssU16                vaporizer;                       // vaporization rate
   ssU16                mild;                            // mild intensity off %
   ssU16                strong;                          // strong intensity off %
   ssU32                persistence;                     // the persistence time
   ssS32                userId;                          // user ID for the callback
   ssSystemId           systemId;                        // system scentId
   ssU32                key[kMaxKeyLength];              // security key
   ssCallback           callback;                        // user callback address
   ssU16                numberOfComponents;              // # of components, 1 based
   ssScentComponent     component[kMaxScentComponents];// scent component(s)
   ssString             reserved[kReservedLength];       // (reserved)
   }ssScent;

typedef ssScent * ssScentPtr;

#define kSizeOfSsScent  sizeof(ssScent)
```

CONFIDENTIAL

Figure 7.0 IScentWareSDK

```
interface IScentWareSDK : IUnknown
   {
   virtual ssBool open(ssVoid)=0;

   virtual ssBool close(ssVoid)=0;

   virtual ssBool setVaporizer(ssU16 vaporizationRate)=0;

   virtual ssBool setDiffuser(ssU16 diffusionRate)=0;

   virtual ssBool setComPort(ssStringPtr portName)=0;

   virtual ssBool setScentStatus(ssBool onOrOff)=0;

   virtual ssBool setDevicePresent(ssBool yesOrNo)=0;

   virtual ssBool setCallback(ssScentPtr scentObject,
                              ssCallback callback,
                              ssS32 userId)=0;

   virtual const char * getVersion(ssVoid)=0;

   virtual const char * getDriverVersion(ssVoid)=0;

   virtual ssErrorStructPtr getErrors(ssVoid)=0;

   virtual ssBool errorExists(ssVoid)=0;

   virtual ssBool isOpen(ssVoid)=0;

   virtual  ssBool isDevicePresent(ssVoid)=0;

   virtual  ssBool isPersistent(ssVoid)=0;

   virtual  ssBool isPersistent(ssScentPtr scentObject)=0;

   virtual ssBool emitScent(  ssScentPtr scentObject,
                              ssIntensity intensity=ssNormal)=0;

   virtual ssBool emitScent(  ssScentPtr scentObject,
                              ssIntensity intensity,
                              ssU16 diffusion,
                              ssU16 vaporization)=0;
   };
```

CONFIDENTIAL

Figure 7.0 Boilerplate Functions

```
IScentWareSDK *CoCreateScentWare(void);

IScentWareSDK *CreateScentWare(void);

ssBool DestroyScentWare(IScentWareSDK *theScentWareSDK);

ssVoid ShowScentWareErrors(IScentWareSDK *theScentWareSDK);

ssStringPtr GetScentWareErrorString(IScentWareSDK *theScentWareSDK);

ssBool PlatformIsScentEnabled(ssVoid);
```

CONFIDENTIAL

SDK Contents

This SDK contains the following items.

iSmellDriver.dll

This is the low level driver for an iSmell, a runtime machine that is equipped with an iSmell device will have this driver on it.

Do not distribute this file (iSmellDriver.dll) with your application. It is for development purposes only.

ScentWareSDK.dll

This is the DLL that will be distributed with your application. It will make use of the *iSmellDriver* DLL if it is present on the runtime machine, or in the absence of the iSmellDriver it will fail gracefully.

ScentWareSDK.h

This is the header file for ScentWare development. Include this in any source file that makes use of ScentWare components.

CreateScentWareSDK.cpp

This is a module of boilerplate code that may be used to quick start the SDK and to display SDK errors.

Test Scents

cinnamon, ocean, greenGrass and smoke. These are authorized scent objects for testing with the virtual iSmell emulation software.

iSmell Emulator.exe and names.txt

The iSmell emulator and its support file names.txt.

CONFIDENTIAL

Glossary

Persistence

The amount of time from the beginning of the emission of the scent until it is no longer subjectively distinguishable in the air.

Emission of Scent

The physical emission of a small quantity of scent essence by the iSmell device.

Diffusion

The amout of air passed throught the scent vaporization chamber. In the SDK this is represented as a number from 0-255, (0==off). Lower diffusion rates will soften the strength of a scent.

Vaporization

The amount of heat applied to the scent essence in the vaporization chamber. This is represented in the SDK as a number from 0-kMaxVaporizerSetting, (0==off). Higher vaporization rates will create scents with a sharp onset and shorter persistence, as the essence will be vaporized more quickly.

Intensity

The subjective strength of a scent from mild to strong. See ssIntensity Fig. 1.4

CONFIDENTIAL

Index

Quickstart 3
Security Key 4
Theory of Operations 5
Code samples and boilerplate 7
Virtual iSmell emulator 8
Library Reference 9
close 10
emitScent 11
emitScent 12
errorExists 13
getDriverVersion 14
getErrors 15
getVersion 16
isDevicePresent 17
isOpen 18
isPersistent 19
isPersistent 20
open 21
setCallback 22
setComPort 23
setDevicePresent 24
setDiffuser 25
setScentStatus 26
setVaporizer 27
Appendix 28
Figure 1.0 Types 29
Figure 2.0 ssBool 30
Figure 3.1 ssSdkError 31
Figure 3.2 ssThreadError 32
Figure 3.3 ssDriverError 33
Figure 3.4 ssCommIoError 34
Figure 3.5 ssUsbIoError 35
Figure 3.6 ssErrorStruct 36
Figure 4.0 ssCallback 37
Figure 5.0 ssIntensity 38
Figure 6.0 ssScent 39
Figure 7.0 iScentWareSDK 40
Figure 7.0 Boilerplate 41
SDK contents 42
Glossary 43

CLAIMS

What is claimed is:

1. An emitting device for generating a scent comprising:

a memory module for receiving a electronically represented scent profile, wherein the scent profile represents the scent to be generated;

a plurality of containers to contain chemical components which may be selectively activated to create a perception of the scent;

a plurality of activating devices for activating individual chemical components; and

a controller which activates the individual chemical components in accordance with the electronically represented scent profile to generate the scent.

2. An emitting device as in claim 1 wherein the activating devices vaporize the individual chemical components.

3. An emitting device as in claim 1 wherein the controller selectively vaporizes the individual chemical components in appropriate relative intensity in accordance with the scent profile.

4. An emitting device as in claim 3 further comprising a mixer for blending the vaporized chemical components.

5. An emitting device as in claim 4 further comprising a pump or pressurized gas to provide a stream of air to transport and emit the vaporized chemical components as the scent emitted from the device.

6. An emitting device as in claim 1 wherein the plurality of containers to contain chemical components are contained in an interchangeable cartridge.

7. An emitting device as in claim 1 wherein the interchangeable cartridge contains a set of chemicals specific to a particular application.

8. An emitting device as in claim 1 wherein an ordered set of elements is used to create the scent profile, and said ordered set of elements is selected from: a set of mammalian olfactory receptors; a set of mammalian olfactory receptor structures modeled on a computer; a set of odorant substances sampled by one or more humans; a set of odorant substance structures modeled on a computer; a set of odorant descriptors; a set of response elements in an artificial scent-sensing device; or a set of odorant substances available in a scent emitting unit.

9. An emitting device as in claim 8 wherein the controller further comprises a computer program to transform a scent profile based on a set of mammalian olfactory receptor structures modeled on a computer, a set of odorant substances sampled by one or more humans, a set of odorant substance structures modeled on a computer, a set of odorant descriptors, or a set of response elements in an artificial scent-sensing device, into a scent profile based on a set of odorant substances available in the emitting device.

10. An emitting device as in claim 1 wherein at least one of the chemical components is in solid phase.

11. An emitting device as in claim 1 wherein at least one of the chemical components is in liquid phase.

12. An emitting device as in claim 11 wherein the liquid phase chemical component is pressurized for delivery via vaporization or as an aerosol.

13. A scent emitting device comprising:

a plurality of reservoirs for containing chemical components;

at least one evaporation chamber in which chemical components evaporate and mix;

a plurality of capillaries in fluid communication with the reservoirs and the evaporation chamber, wherein the chemical components flow through the capillaries to the evaporation chamber by capillary action; and

a control device for selectively regulating the flow of the chemical component through each capillary, so as to deliver the appropriate chemical components in the appropriate amount to the evaporation chamber to be evaporated and mixed to create a desired scent.

14. A scent emitting device as in claim 13 wherein the control device comprises:

micro-valves, each regulating the flow through a capillary; and

a controller structured and configured to selectively control the operation of the micro-valves based on input data that represents the scent to be created.

15. A scent emitting device as in claim 13 wherein the evaporation chambers comprises heating elements which heat the chemical components delivered to the evaporation chamber.

16. A scent emitting device as in claim 13 further comprising a pump or pressurized gas to provide a stream of air through the evaporation chamber to transport and emit the evaporated chemical components as the scent emitted from the device.

17. A scent emitting device comprising:

a plurality of reservoirs for containing chemical components;

an evaporation platform on which chemical components are deposited, evaporated and mixed;

a plurality of conduits in fluid communication with the reservoir and the evaporation platform, wherein the chemical components flow through the conduits to the evaporation platform; and

heating elements which selectively heat the chemical component in each reservoir to facilitate delivery of the chemical component through the conduit and deposition on the evaporation platform; and

a control device for selectively regulating the flow of the chemical component through each conduit by selectively controlling heating of the heating means, so as to deliver the appropriate chemical components in the appropriate amount to the evaporation platform to be evaporated and mixed to create a desired scent.

18. A scent emitting device as in claim 17 further comprising heating elements to heat the chemical components deposited on the evaporation platform.

19. A scent emitting device as in claim 18 wherein the chemical components are deposited on respective regions on the evaporation platform and the heating elements which heat the chemical components deposited selectively heat the regions to achieve a desired evaporation rate for the respective chemical components, thereby achieving the desired intensity of the chemical components for mixing to create the scent.

20. A scent emitting device as in claim 19 wherein the control device synchronizes the operations of the separate heating elements with input data that represents the scent to be created.

21. A scent emitting device as in claim 20 further comprising a pump or pressurized gas to provide a stream of air through the evaporation chamber to transport and emit the evaporated chemical components as the scent emitted from the device.

22. A scent emitting device as in claim 17 wherein the evaporation platform is movable so as to transport the chemical components deposited thereon away from the conduits after scent emission.

23. A scent emitting device as in claim 22 wherein the evaporation platform transports the chemical components to a heating station downstream in the direction of travel of the evaporation platform, whereby the chemical components are selectively heated and evaporated at the heating station so as to minimize evaporation outside of the heating station.

24. A scent emitting device as in claim 22 wherein the evaporation platform is made of a fibrous sheet material which absorbs the chemical components deposited thereon.

25. A scent emitting device as in claim 24 wherein the fibrous sheet material is blotting paper.

26. A scent emitting device as in claim 24 wherein the fibrous sheet material is in the form of a continuous roll which is unwound from a first roller, the unwound portion of said sheet passing under the conduits and to the heating station, and then wound at a second roller to wind back up the unwound portion after it passes the heating station.

27. A scent emitting device as in claim 22 wherein the heating elements, the heating station and the movable evaporation platform are synchronized so as to deposit the appropriate chemical components in the appropriate amounts and evaporate them in the appropriate intensities, based on input data that represents the scent to be emitted.

28. A scent emitting device as in claim 17 wherein the evaporation platform comprises a plurality of pads, each positioned under a conduit.

29. A scent emitting device as in claim 28 further comprising heating elements for selectively heating the pads.

30. A scent emitting device comprising:

a plurality of reservoirs for containing chemical components;

an air flow tube operatively coupled to each reservoir;

a plurality of conduits in fluid communication with the reservoirs and the air flow tube, in a manner wherein the chemical components are drawn through the conduits by air flow in the air flow tube; and

a control device for selectively regulating the flow of the chemical components through each conduit, so as to deliver the appropriate chemical components in the appropriate amount to the air flow tube.

31. A scent emitting device as in claim 30 wherein the control device comprises:

a plurality of valves, each regulating the flow through a conduit; and

a controller structured and configured to selectively control the operation of the valves based on input data that represents the scent to be created.

32. A scent emitting device as in claim 30 further comprising a pump or pressurized gas to provide a stream of air through the evaporation chamber to transport and emit the evaporated chemical components as the scent emitted from the device.

33. A scent emitting device as in claim 32 wherein the air flow tube is a Venturi tube.

34. A scent emitting device as in claim 30 wherein the conduits comprise tubes.

35. A scent emitting device as in claim 30 wherein the conduits comprise channels through a high surface area medium.

36. A scent emitting device as in claim 35 wherein the high surface area medium comprises a gauze material.

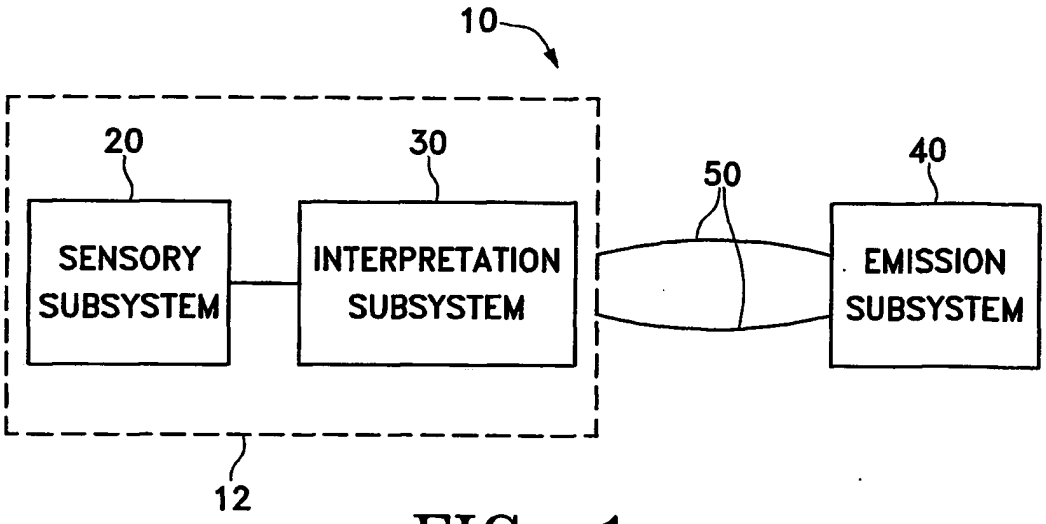

10
20
30
50
40
SENSORY SUBSYSTEM
INTERPRETATION SUBSYSTEM
EMISSION SUBSYSTEM
12

FIG. 1

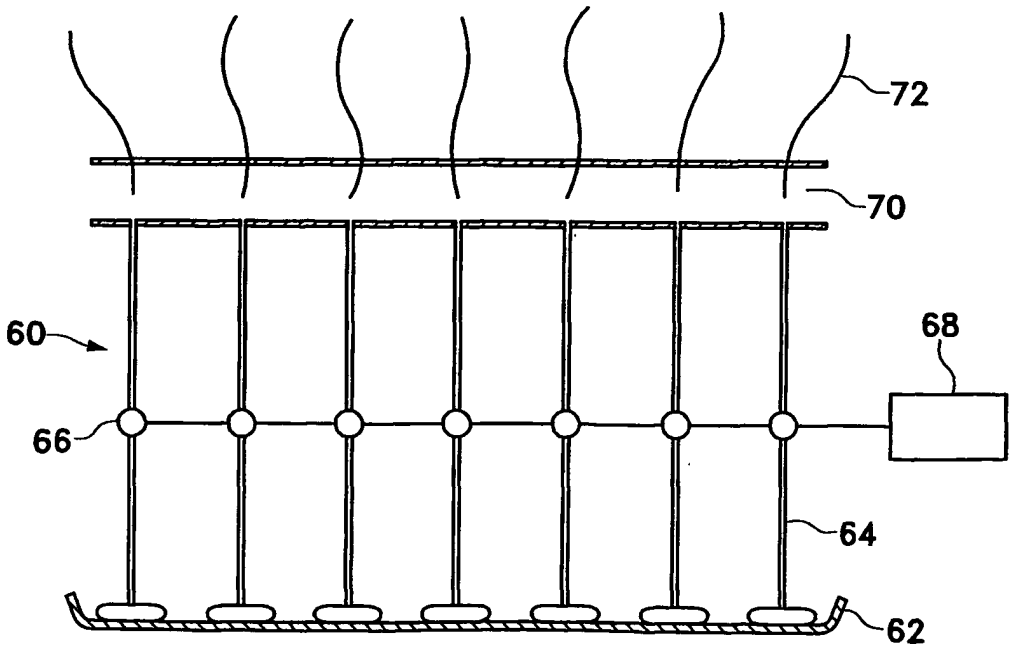

72
70
60
68
66
64
62

FIG. 2

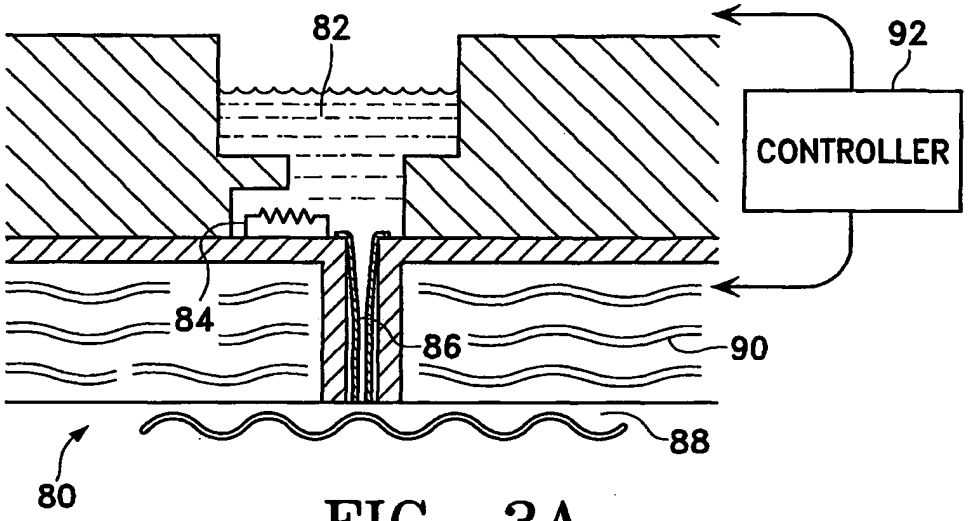

82
92
CONTROLLER
84
86
90
88
80

FIG. 3A

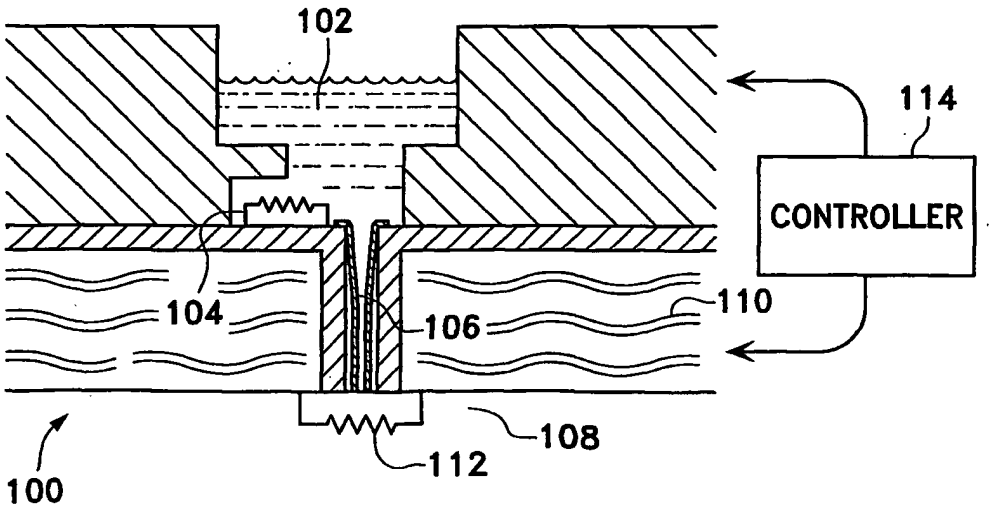

102
114
CONTROLLER
104
106
110
108
112
100

FIG. 3B

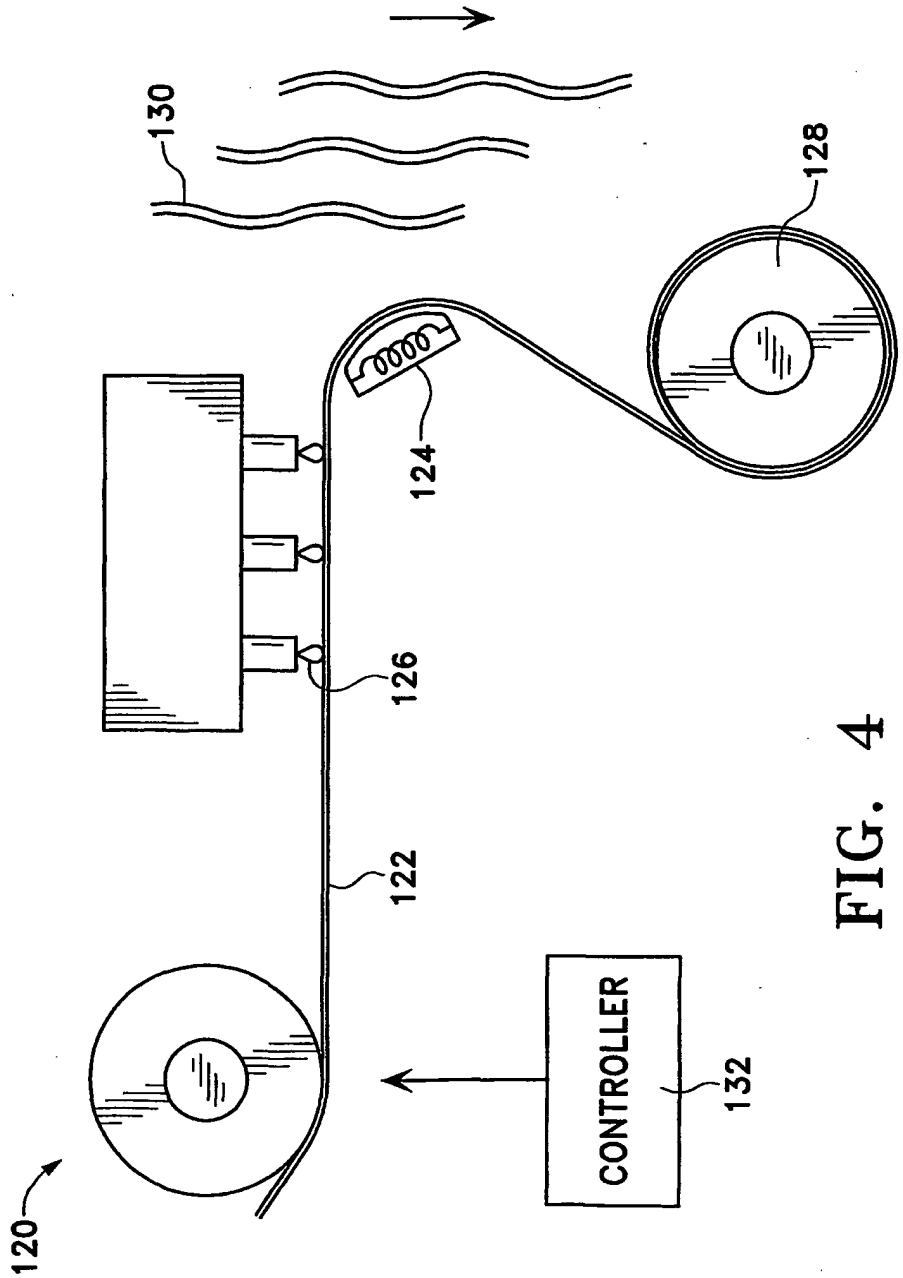


FIG. 4

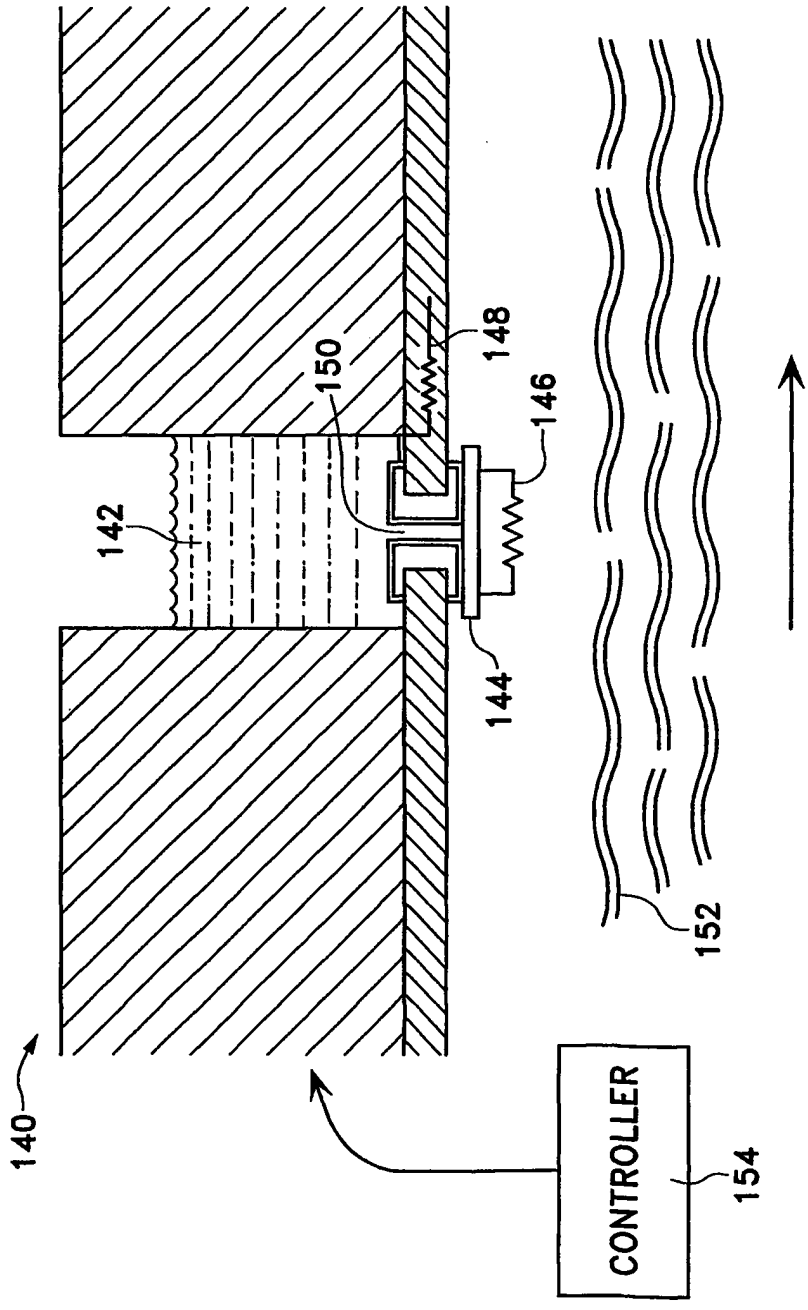


FIG. 5

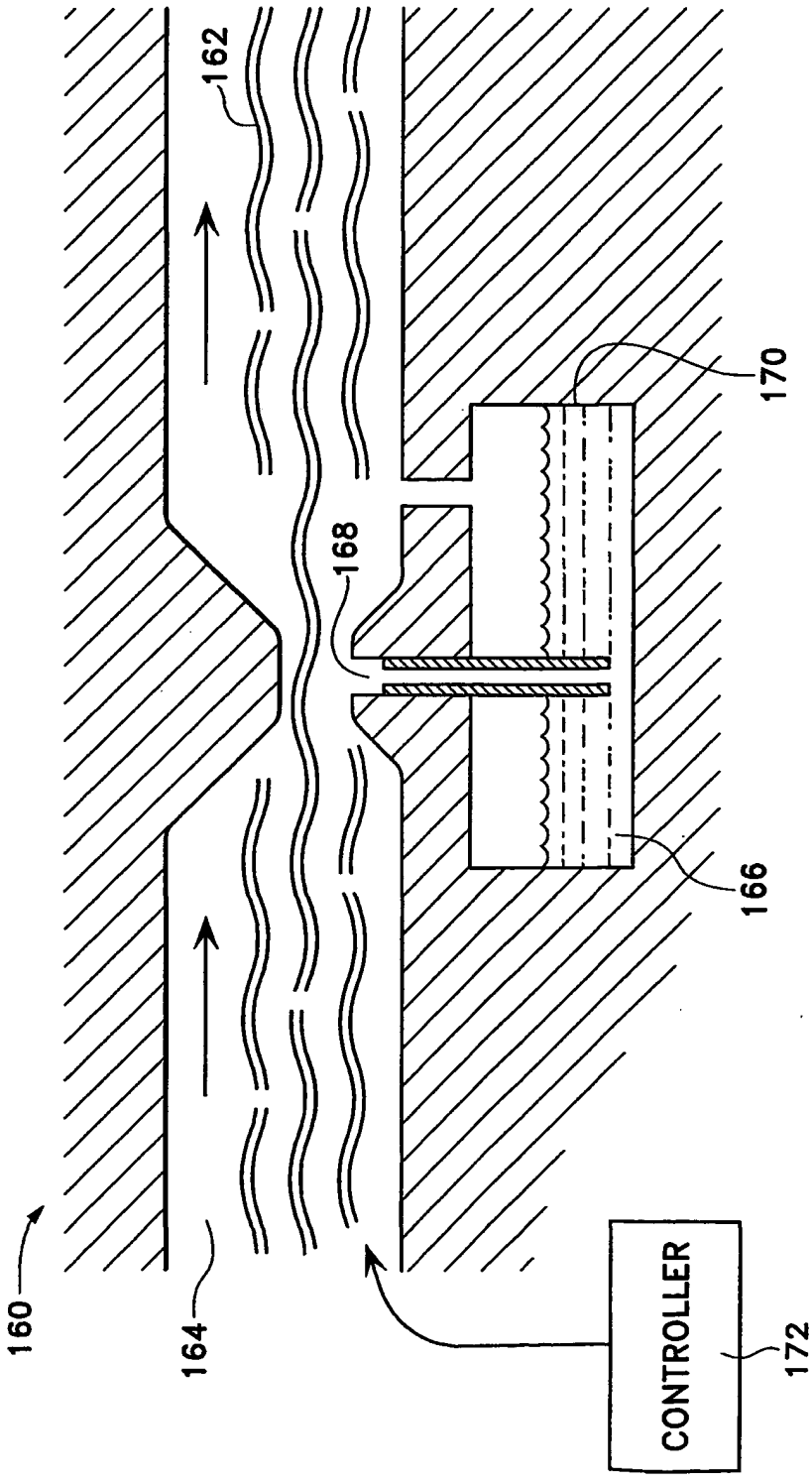


FIG. 6

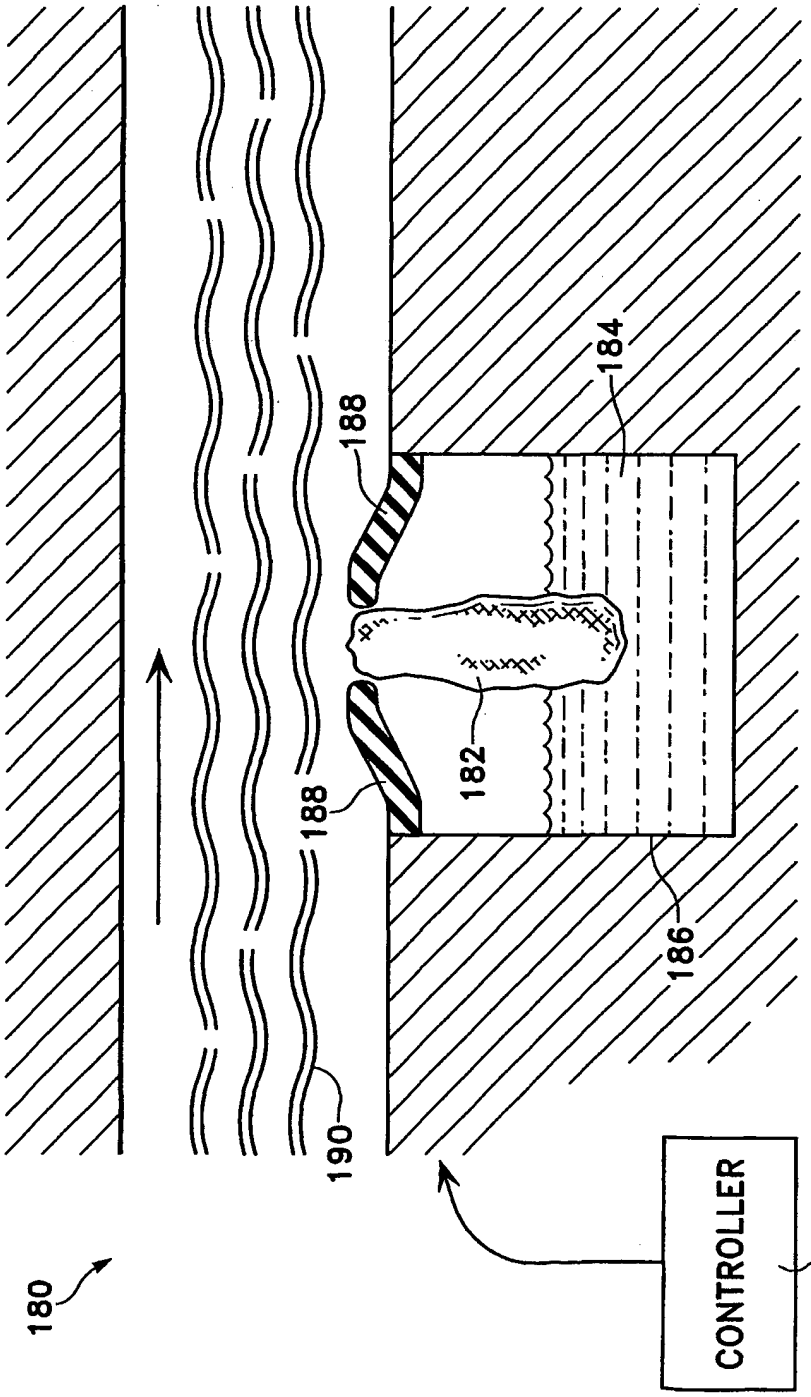


FIG. 7

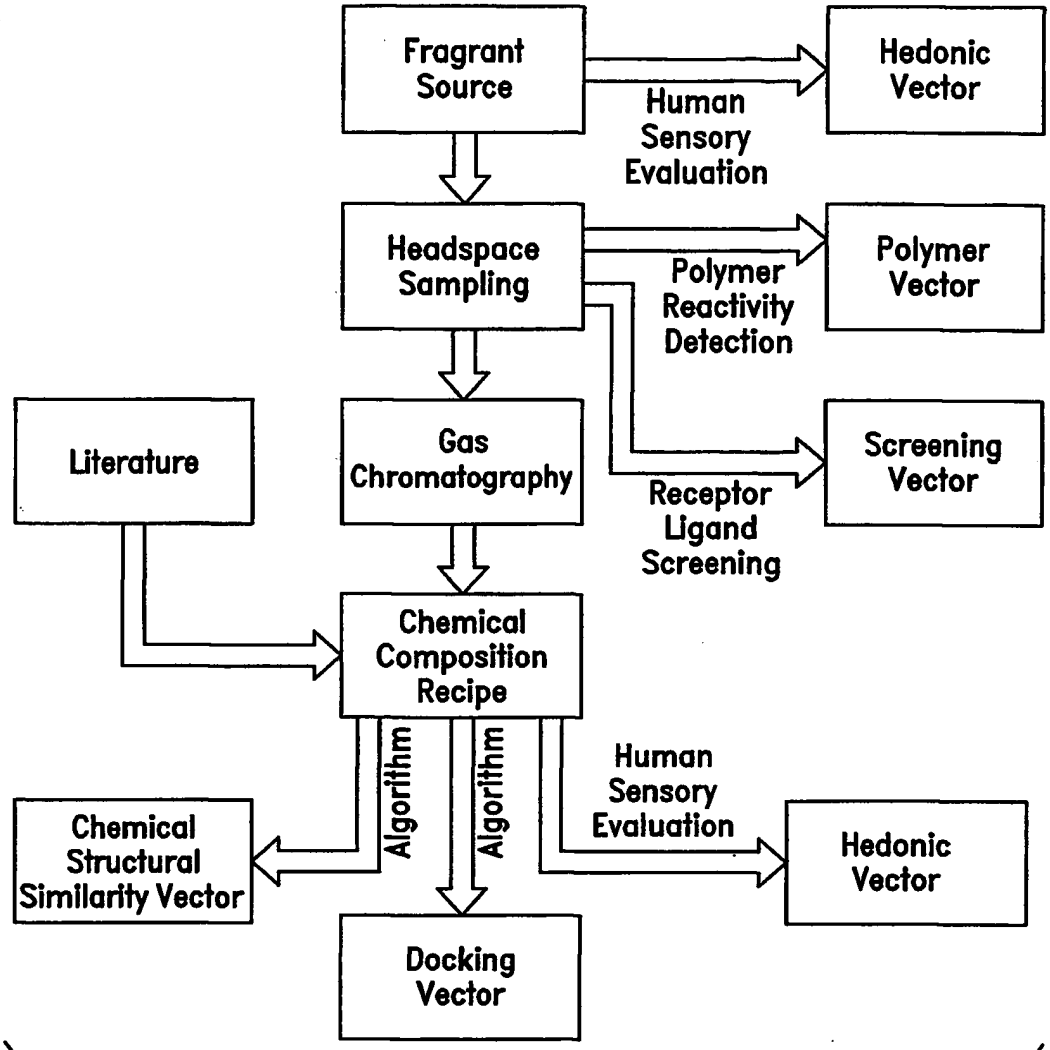

Fragrant Source
Hedonic Vector
Human Sensory Evaluation
Headspace Sampling
Polymer Vector
Polymer Reactivity Detection
Literature
Gas Chromatography
Screening Vector
Receptor Ligand Screening
Chemical Composition Recipe
Algorithm
Algorithm
Human Sensory Evaluation
Chemical Structural Similarity Vector
Hedonic Vector
Docking Vector

FIG. 8A FIG. 8B

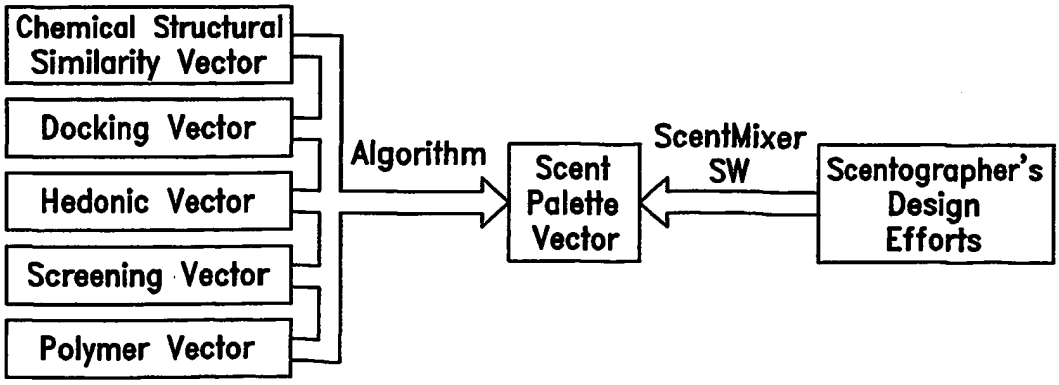
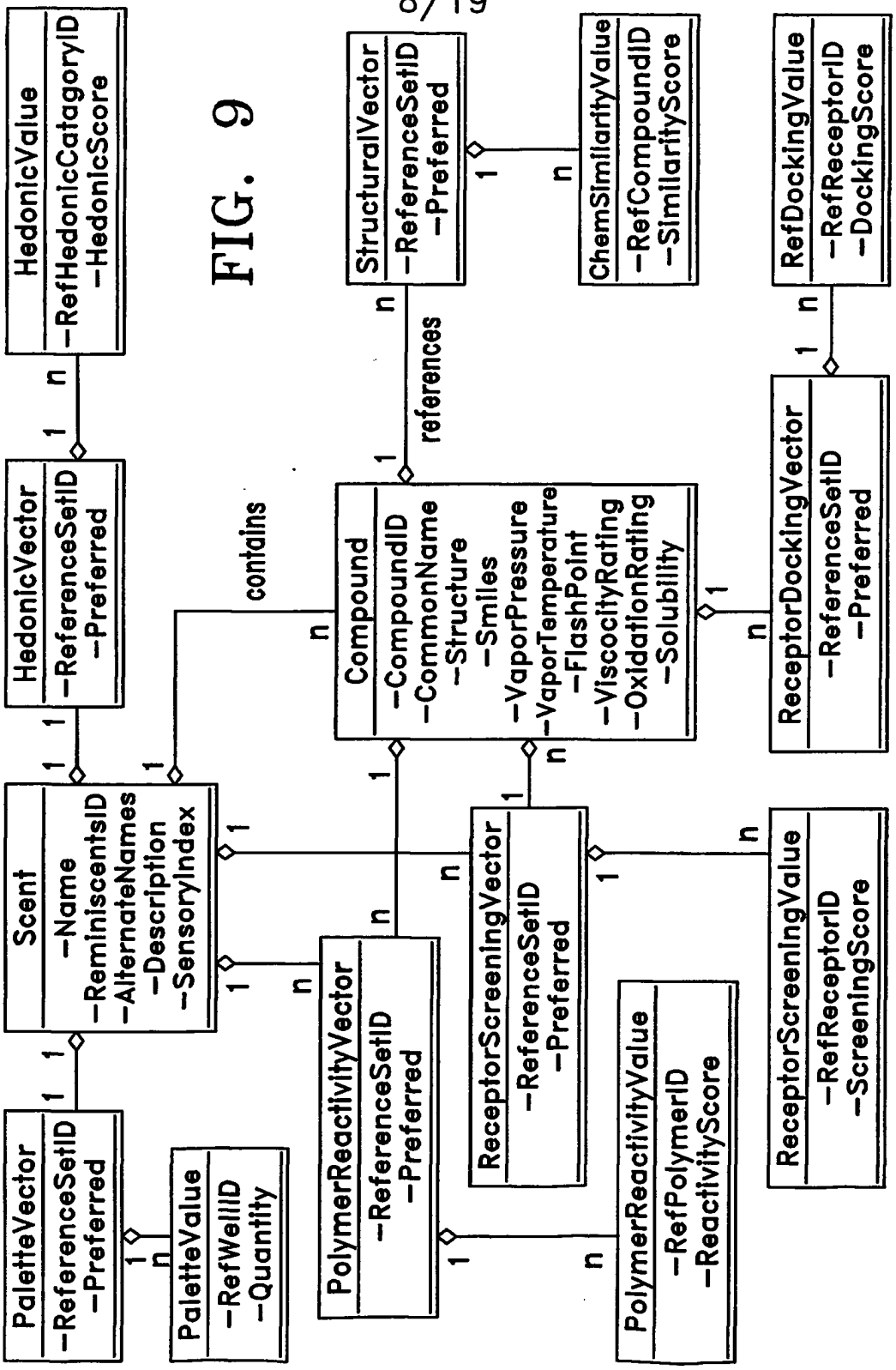

Chemical Structural Similarity Vector
Docking Vector
Hedonic Vector
Screening Vector
Polymer Vector
Algorithm
Scent Palette Vector
ScentMixer SW
Scentographer's Design Efforts

FIG. 9


HedonicValue
-RefHedonicCatagoryID
-HedonicScore
HedonicVector
-ReferenceSetID
-Preferred
Scent
-Name
-ReminiscentsID
-AlternateNames
-Description
-SensoryIndex
PaletteVector
-ReferenceSetID
-Preferred
PaletteValue
-RefWellID
-Quantity
contains
Compound
-CompoundID
-CommonName
-Structure
-Smiles
-VaporPressure
-VaporTemperature
-FlashPoint
-ViscocityRating
-OxidationRating
-Solubility
references
StructuralVector
-ReferenceSetID
-Preferred
ChemSimilarityValue
-RefCompoundID
-SimilarityScore
RefDockingValue
-RefReceptorID
-DockingScore
ReceptorDockingVector
-ReferenceSetID
-Preferred
PolymerReactivityVector
-ReferenceSetID
-Preferred
PolymerReactivityValue
-RefPolymerID
-ReactivityScore
ReceptorScreeningVector
-ReferenceSetID
-Preferred
ReceptorScreeningValue
-RefReceptorID
-ScreeningScore

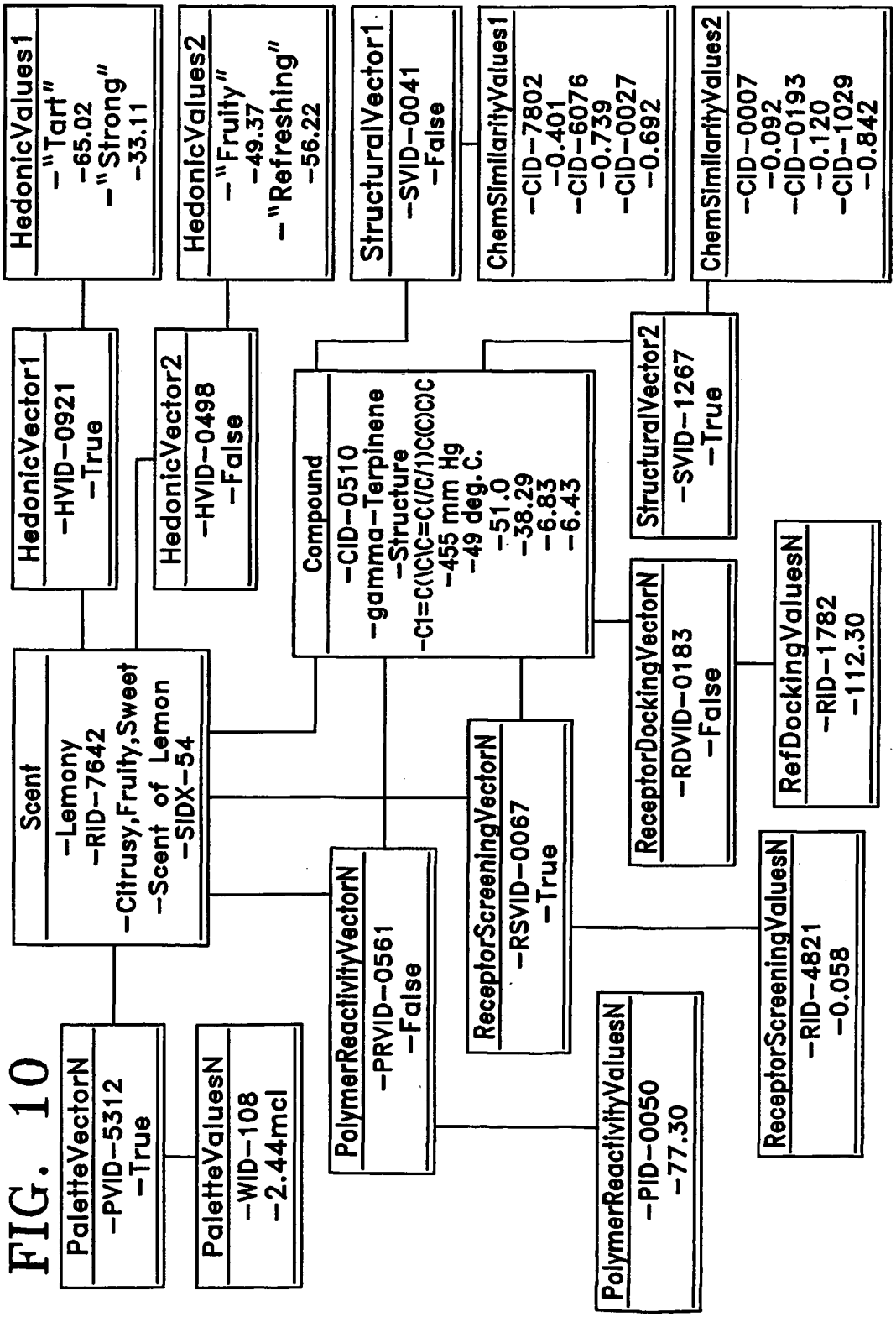

HedonicValues1
-"Tart"
-65.02
-"Strong"
-33.11
HedonicValues2
-"Fruity"
-49.37
-"Refreshing"
-56.22
StructuralVector1
-SVID-0041
-False
ChemSimilarityValues1
-CID-7802
-0.401
-CID-6076
-0.739
-CID-0027
-0.692
ChemSimilarityValues2
-CID-0007
-0.092
-CID-0193
-0.120
-CID-1029
-0.842
HedonicVector1
-HVID-0921
-True
HedonicVector2
-HVID-0498
-False
Compound
-CID-0510
-gamma-Terpinene
-Structure
-C1=C(\C\C=C(/C/1)C(C)C)C
-455 mm Hg
-49 deg.C.
-51.0
-38.29
-6.83
-6.43
StructuralVector2
-SVID-1267
-True
Scent
-Lemony
-RID-7642
-Citrusy,Fruity,Sweet
-Scent of Lemon
-SIDX-54
ReceptorDockingVectorN
-RDVID-0183
-False
RefDockingValuesN
-RID-1782
-112.30
ReceptorScreeningVectorN
-RSVID-0067
-True
PolymerReactivityVectorN
-PRVID-0561
-False
ReceptorScreeningValuesN
-RID-4821
-0.058
PaletteVectorN
-PVID-5312
-True
PaletteValuesN
-WID-108
-2.44mcl
PolymerReactivityValuesN
-PID-0050
-77.30

FIG. 10

Name:	"Lemony"
ReminiScentsID:	RID-7642
Description:	Scent of Lemon
AlternateNames:	Citrusy, Fruity, Sweet
Sensory Index:	SIDX-54:

Hedonic Vectors

preferred	Vec. ID	Hed. Category ID	Value
	763	12("Tart")	0.487
	763	15("Strong")	0.521
	763	17("...")	0.613
	763	9	0.494
	763	3	0.725
.	.	.	.
.	.	.	.
.	.	.	.
	211	77("Lemony")	0.161
	211	61("Fruity")	0.540
	211	5("...")	0.124
	211	83	0.374
	211	39	0.881
.	.	.	.
.	.	.	.
.	.	.	.

FIG. 11A

Structure

Name:	gamma-Terpinene
SMILES:	C1=C(\C\C=C(/C/1)C(C)C)C
Vap. Press.:	
Vap. Temp.:	
Flash Point:	51 deg. C.
Viscocity Index:	
Oxidation Rating:	
Solubility:	

Chemical Similarity Vectors

preferred	Vec. ID	Cmpd. ID	Value
	123	7	0.587
	123	193	0.731
	123	1029	0.683
	123	869	0.474
	123	1297	0.722
.	.	.	.
.	.	.	.
.	.	.	.
	456	7802	0.167
	456	29	0.560
	456	5031	0.129
	456	1239	0.384
	456	297	0.231
.	.	.	.
.	.	.	.
.	.	.	.

FIG. 11B

Docking Receptor Vectors

preferred	Vec. ID	Receptor ID	Value
	355	1255	0.587
	355	93	0.731
	355	1729	0.683
	355	821	0.474
	355	1597	0.722
.	.	.	.
.	.	.	.
.	.	.	.
	921	7842	0.167
	921	439	0.560
	921	531	0.129
	921	4129	0.384
	921	797	0.231
.	.	.	.
.	.	.	.
.	.	.	.

Additional Vectors:

Pallette Vectors

Polymer Reactivity Vectors

Receptor Screening Vectors

FIG. 11C

FIG. 12

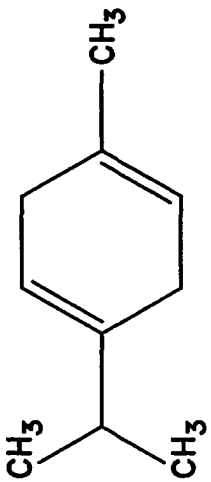
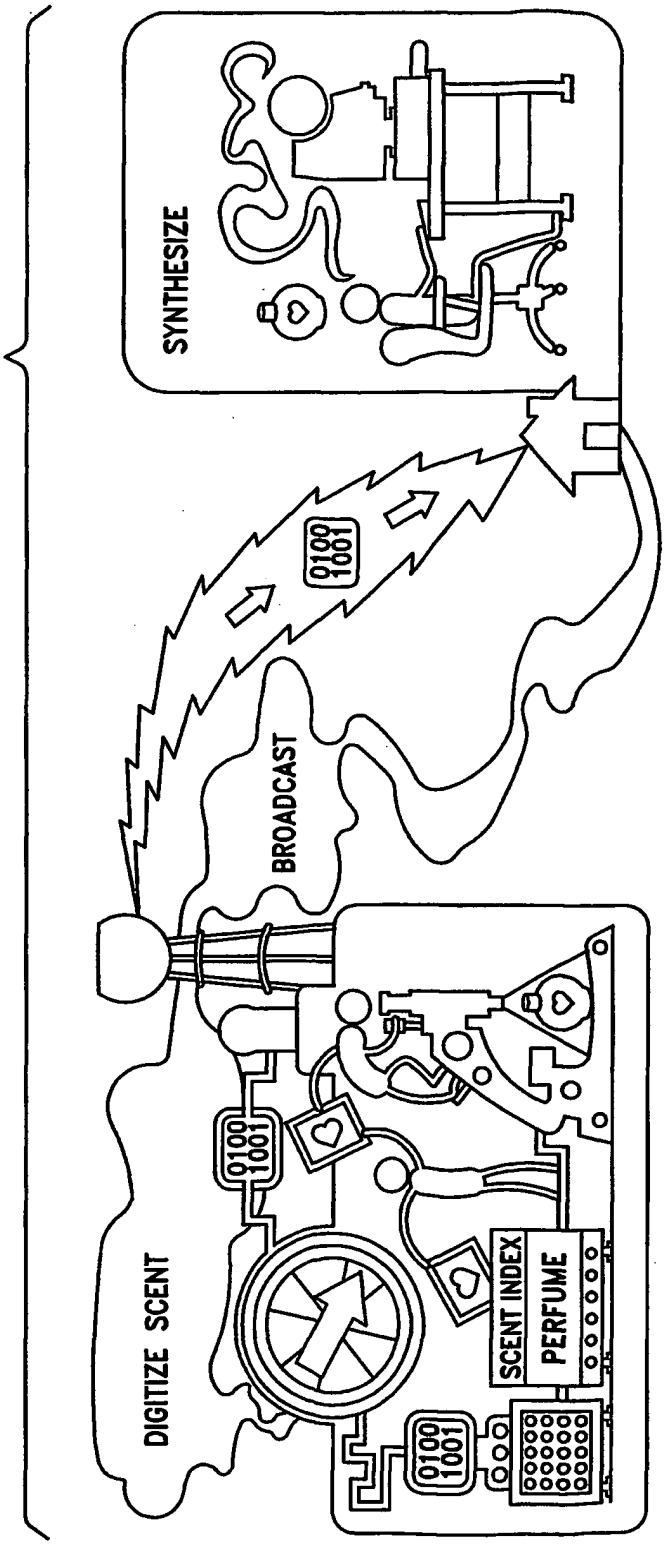

CH3
CH3
CH3

FIG. 13


SYNTHESIZE
0100
1001
BROADCAST
0100
1001
DIGITIZE SCENT
SCENT INDEX
PERFUME
0100
1001

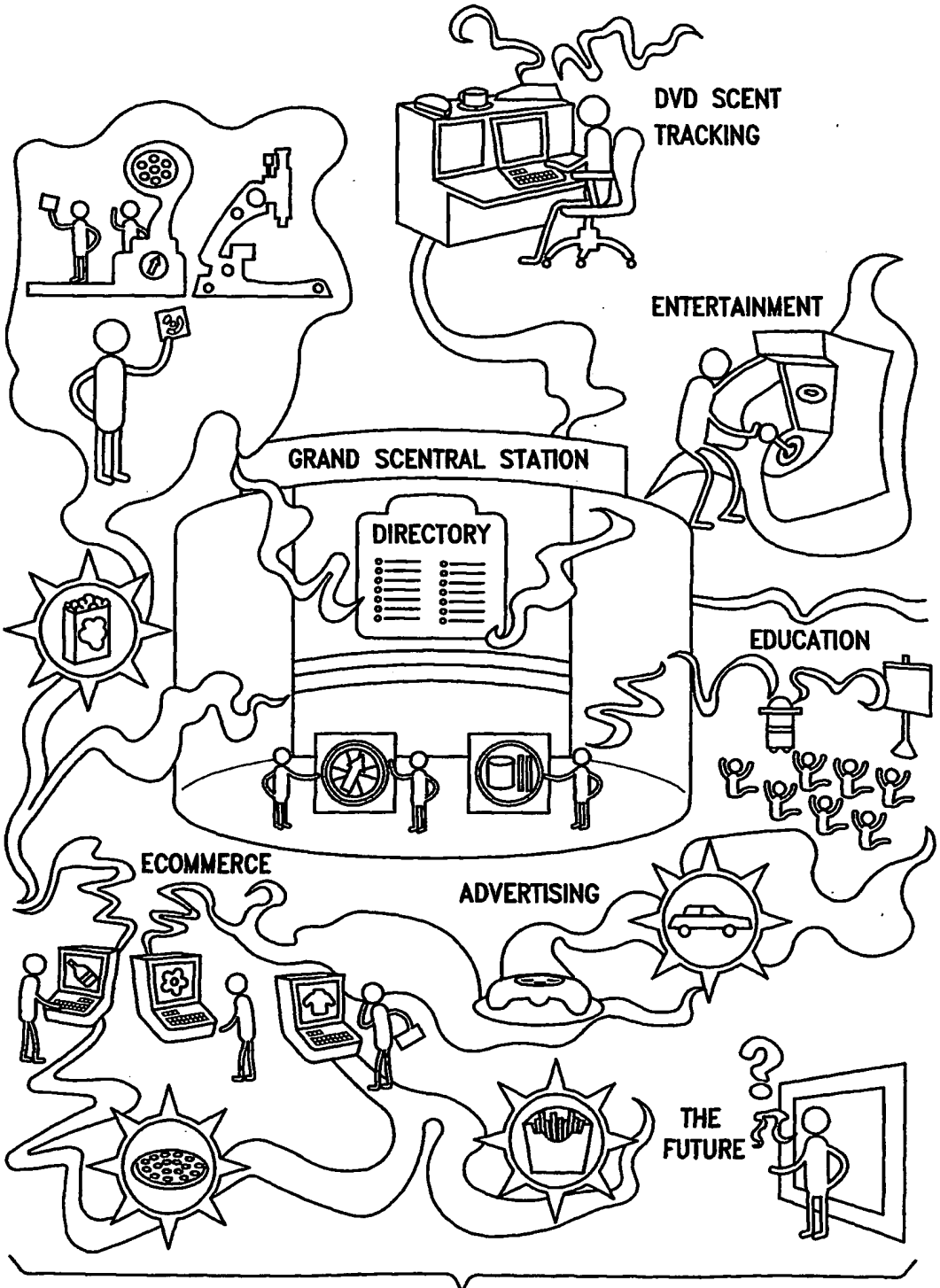

DVD SCENT
TRACKING
ENTERTAINMENT
GRAND SCENTRAL STATION
DIRECTORY
EDUCATION
ECOMMERCE
ADVERTISING
THE
FUTURE

FIG. 14

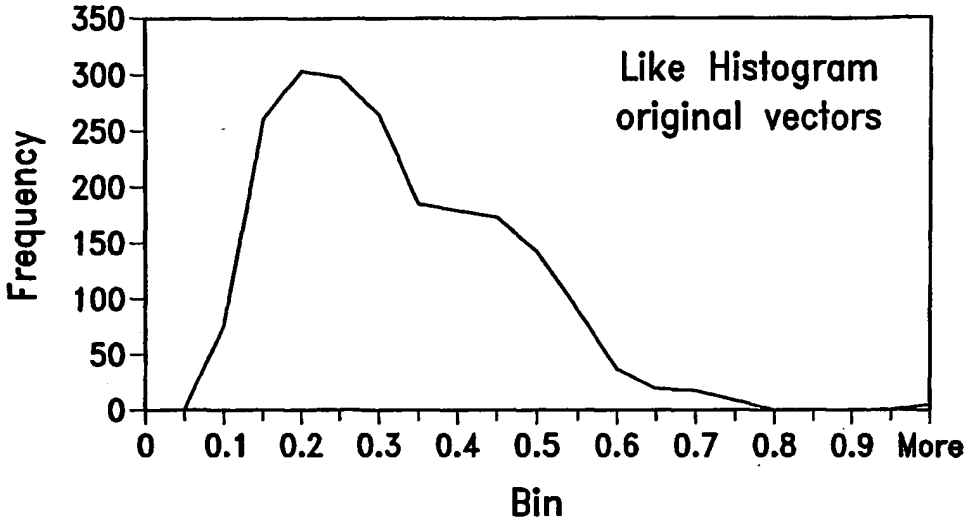

Like Histogram
original vectors
Frequency
350
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 15

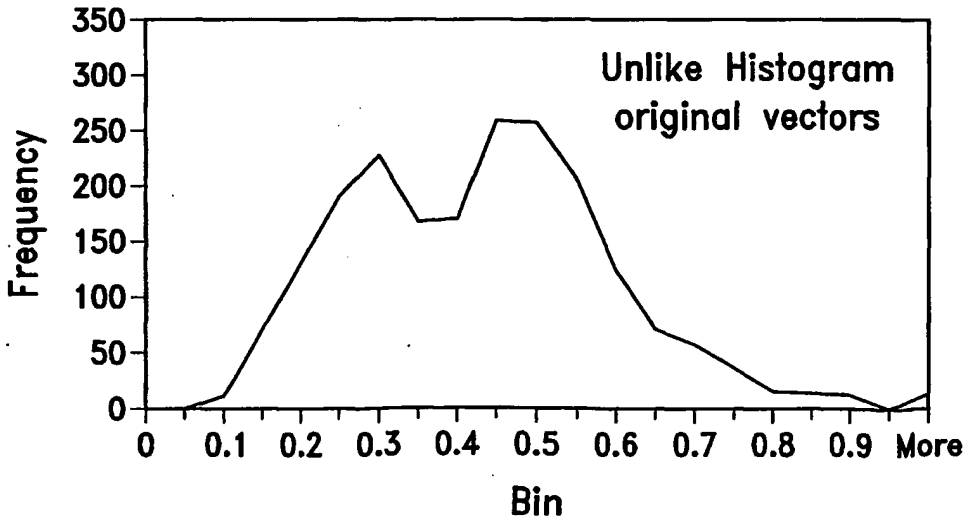

Unlike Histogram
original vectors
Frequency
350
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 16

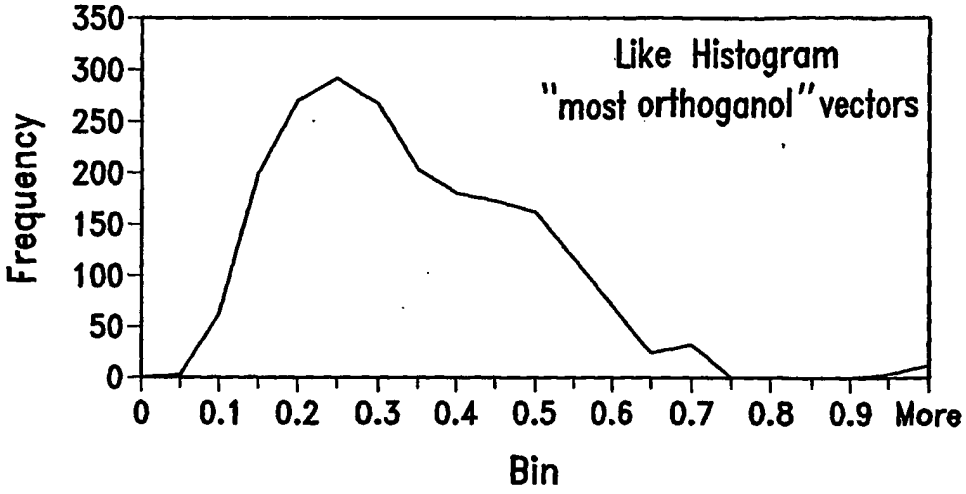

Like Histogram
"most orthoganol" vectors
Frequency
350
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 17

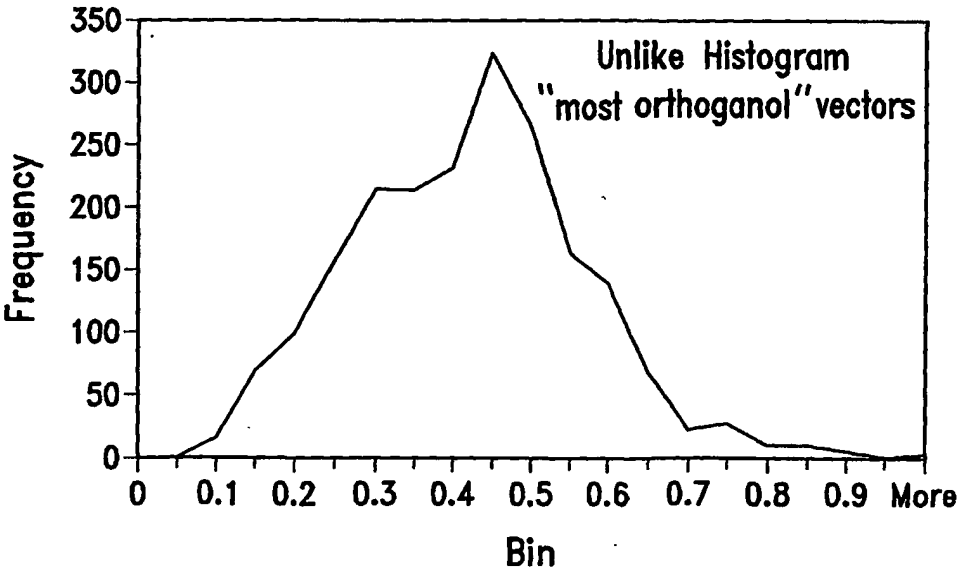

Unlike Histogram
"most orthoganol" vectors
Frequency
350
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 18

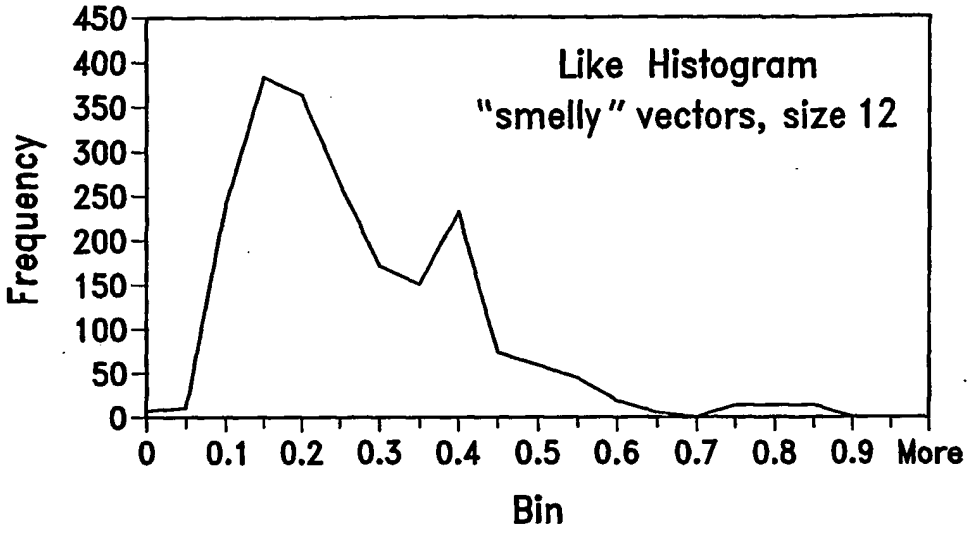

Like Histogram
"smelly" vectors, size 12
Frequency
450
400
350
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 19

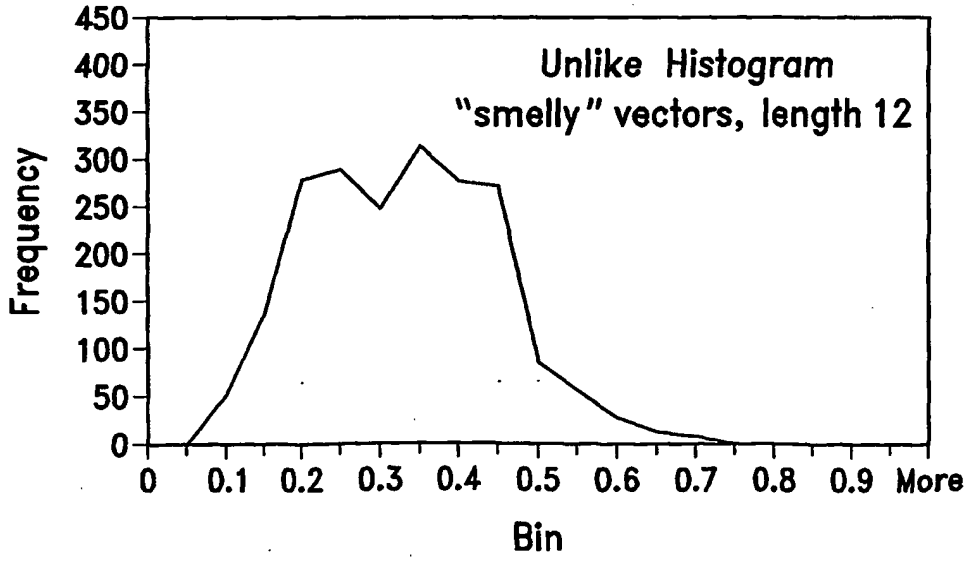

Unlike Histogram
"smelly" vectors, length 12
Frequency
450
400
350
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 20

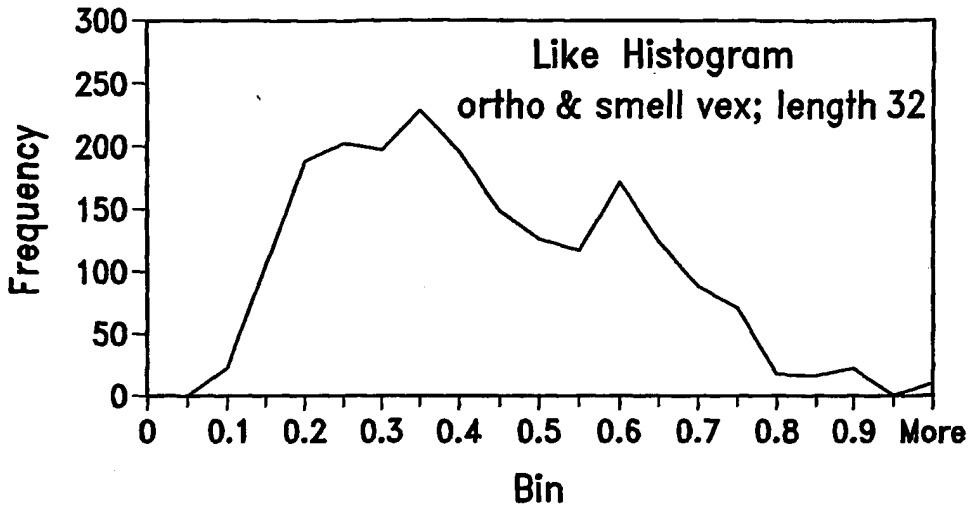

Like Histogram
ortho & smell vex; length 32
Frequency
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 21

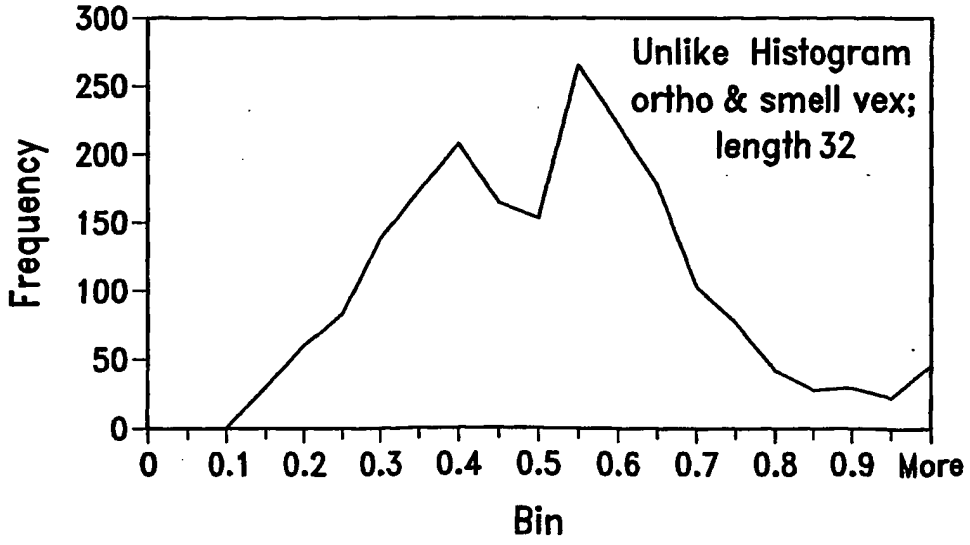

Unlike Histogram
ortho & smell vex;
length 32
Frequency
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 22

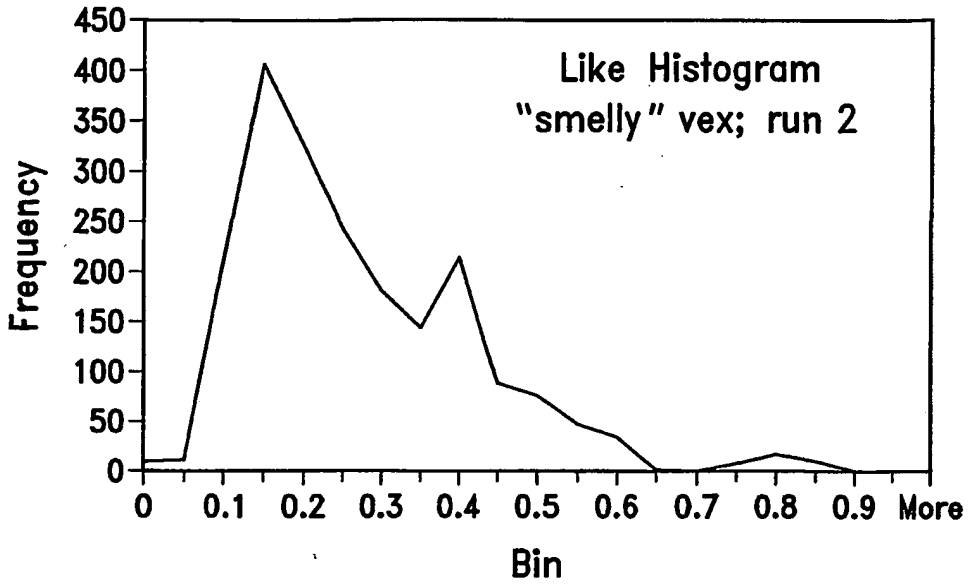

Like Histogram
"smelly" vex; run 2
Frequency
450
400
350
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 23

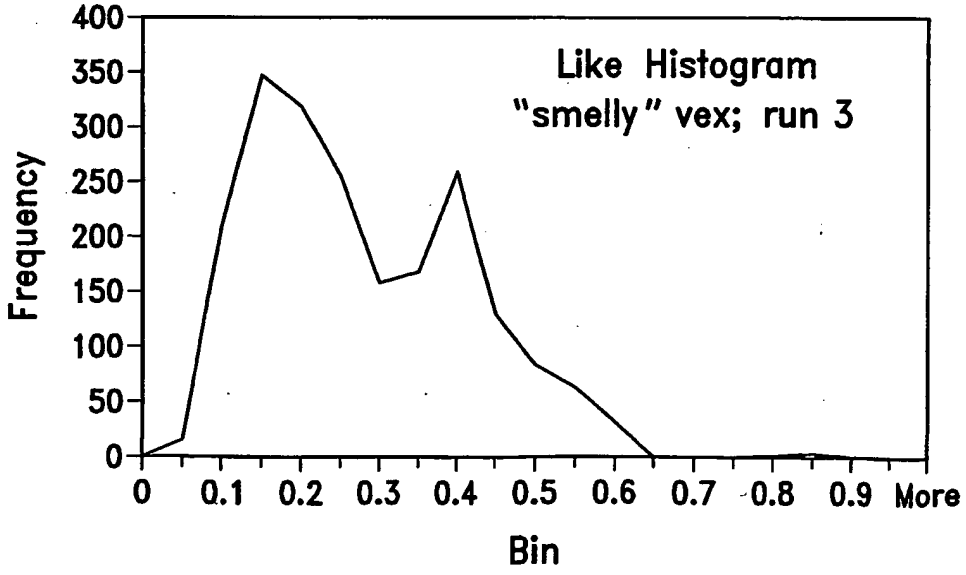

Like Histogram
"smelly" vex; run 3
Frequency
400
350
300
250
200
150
100
50
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 More
Bin

FIG. 24

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/20029

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 A61L9/00 A61L9/03 A61L9/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 A61L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC, COMPENDEX, BIOSIS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 99 16476 A (PEREX AGORRETA FRANCISCO JAVIE) 8 April 1999 (1999-04-08) cited in the application the whole document	1-12
X	EP 0 831 384 A (RICOH KK) 25 March 1998 (1998-03-25) cited in the application column 1, line 13 -column 2, line 4 column 4, line 31 -column 5, line 27 column 7, line 13 -column 9, line 55 figures 1,6-14	1-12, 17-29

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 December 2000	02/01/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Muñoz, M

Form PCT/ISA/210 (second sheet) (July 1992)